Environment Canada Environnement Canada

Canada

NABIODAT, THE BIOLOGICAL DATA MANAGEMENT SYSTEM

IWD-AR-WQB-85-90

INLAND WATERS DIRECTORATE
ATLANTIC REGION
WATER QUALITY BRANCH
MONCTON, NEW BRUNSWICK

NABIODAT, The Biological Data Management System

IWD-AR-WQB-85-90

Inland Waters Directorate
Environment Canada

Geoffery Howell
Lawrence Wong
John Fenton
Richard Bingham

Water Quality Branch
Atlantic Region
Moncton, New Brunswick

TABLE OF CONTENTS

	PAGE
Abstract	ii
Resume	ii
1.0 Introduction	1
2.0 System Framework	2
2.1 Species Code	2
2.2 Method, Parameter and Special Purpose Codes	2
3.0 NABIODAT	4
3.1 BIOSHEET	4
3.2 PRTSPEC	4
3.3 PRTCODES	7
3.4 BLDSPEC	7
3.5 BIODAT	11
4.0 System Integrity	20
5.0 Program Listing	21

ABSTRACT

This report gives an introduction to NABIODAT, the Water Quality Branch, Atlantic Region biological data management system. The theoretical basis for the systemand a brief overview and example output for each program are presented. This report also includes the NABIODAT program listing.

RÉSUMÉ

Le présent rapport introduit NABIODAT, le système de gestion des données biologiques, développé par la Direction de la qualité des eaux, région de l'Atlantique. Les bases théoriques du système, un bref résumé et des exemples de rapport d'échantillons pour chaque programmes sont présentés. De plus, ce rapport inclu une copie du logiciel du système NABIODAT.

1.0 Introduction

The majority of Water Quality Branch sampling and analytical effort has in the past been restricted to water chemistry concerns and issues. However, some programs such as LRTAP and Toxic Chemicals have emphasized the need to consider biological parameters in addition to water quality parameters. To date, even though the importance of biological monitoring has been recognized, most of the biological sampling effort has been restricted to synoptic surveys.

A major drawback with incorporating a comprehensive biological component into Water Quality Branch monitoring and survey programs, is the inability to store, manipulate and retrieve all types of biological data within the framework of the NAQUADAT system. This has in many cases resulted in an ad hoc approach to design, data management and interpretation of the biological aspects of Water Quality Branch programs.

The Water Quality Branch, Atlantic Region, is presently incorporating biological components into the regional LRTAP and Toxic Chemical programs. In order to enable adequate management and interpretation of these data, a biological data management has been designed. This system, NABIODAT, has been designed to be independent of NAQUADAT, although provisions have been made to allow cross-referencing with water quality data. This report presents the theoretical basis and source code for the NABIODAT system.

2.0 System Framework

The foundation of any flexible biological data management system is the development of a coding structure that will allow management of data concerning species, sampling methodology and analytical measurements. The NABIODAT system is designed around a theoretical framework of four code types, all of which have a six-character, alpha-numeric format. These code types include species codes, method codes, parameter codes and special purpose codes.

2.1 Species Code

The species code has been designed with a hierarchial taxonomic structure and provides four levels of information. The first digit of the code must be a letter and is used to indicate a phyllum or a collective taxonomic group. The second digit can be alpha-numeric and indicates the class or division within the phyllum. These two digits may not always follow strict taxonomic designation but may instead be defined as a biologically significant group. An example of this would be a zooplankton category which includes both arthropoda and rotifera. The third and fourth characters indicate the genus while the last two digits are reserved for species or sub-species designation. This coding system allows for a theoretical maximum of 23 collective groups, 34 classes/phylla, 1156 genera/class and 1156 species/genera.

2.2 Method, Parameter and Special Purpose Codes

The method, parameter and special purpose codes are differentiated from species codes by the use of a reserved letter "M". The method code commences with the letter "M" and is followed by a letter indicating the overall taxonomic group for which the method applies. The final four digits can be alpha-numeric and provide the flexibility to code for a large number of sampling methods. These codes are also used to provide additional information related to sampling such as equipment dimensions and preservation techniques. The power of the method code is improved by the fact that it has been designed as a variable and thus can take a value. An example of the method code is "ML0104" which indicates a monofilament gill net. The mesh size of the net can be entered as the value.

The parameter code is similar to the method code in that both codes are specific to overall taxonomic groups. The parameter code begins with "MM" and is then followed by the phyllum group letter. These codes are analagous to the parameter codes in NAQUADAT and indicate the analytical or measurement technique employed. An example of a parameter code is "MML003" which represents the number of individuals per sample.

The fourth and final code types are the special purpose codes which commence with "MMM". These codes provide information which refers to the sample as a whole and is not specific to any given taxa. Thus these codes are used to convey data concerning sampling duration and depth as well as to cross-link samples with other samples in either NABIODAT or NAQUADAT. An example of a special purpose code is "MMM101" which indicates the sample depth range (Z1 - Z2 meters).

3.0 NABIODAT

NABIODAT was programmed on a VAX 11/750 using BASIC 2.2 and thus because of the use of RMS indexed files is only transferable to systems with BAS/C+Z or VAX BASIC. In addition, the system does employ some hardware specific control sequences which will require modification on foreign systems. The total data management system is made up of one major program "BIODAT" and a series of four peripheral programs (BIOSHEET, PRTSPEC, PRTCODES, BLDSPEC). All the programs are menu driven and have been designed for a "VT100" terminal.

3.1 BIOSHEET

This program is designed to produce two different data coding sheets. Field sheets can be printed for a series of sampling stations and provide a standardized format for recording field observations. After biological sampling information has been initialized in the main "BIODAT" program, BIOSHEET can also be used to print out a standardized form for results of contracted analyses. This form displays any field information that has been entered onto the system and is designed to facilitate addition of contractor results to the data record. Examples of the field and contractor sheets are presented in Tables 1 and 2 respectively.

3.2 PRTSPEC

Table I

BIOLOGICAL FIELD SHEET

STATION # 01NF02XB0002 SAMPLE # _____

ST.AUGUSTIN R. HEADWATER L.

DATE OF SAMPLING: DAY: ___ MONTH: ___ YEAR: ___

TIME OF SAMPLING: HOUR: ___ MINUTES: ___ ZONE: ___

LENGTH OF TIME SAMPLING: DAYS: ___ HOURS: ___ MINUTES: ___

SUBMITTER ID _____ SAMPLED BY _____

SPECIES	SPECIES CODE	ID #	PARAMETER CODE	PARAMETER VALUE	FIELD METHOD CODE	FIELD METHOD VALUE

ADDITIONAL INFORMATION _____

Table 2

BIOLOGICAL CONTRACTOR'S DATA SHEET

STATION # 01NF02XB0002 SAMPLE # 100000

ST.AUGUSTIN R. HEADWATER L.

DATE OF SAMPLING: DAY: 14 MONTH: SEP YEAR: 1983

TIME OF SAMPLING: HOUR: 12 MINUTES: 00 ZONE: AST

LENGTH OF TIME SAMPLING: DAYS: ___ HOURS: ___ MINUTES: ___

SUBMITTER ID 0085 TESTED BY _____

SPECIES	SPECIES CODE	ID #	PARAMETER CODE	PARAMETER VALUE	DESCRIPTION

ADDITIONAL INFORMATION _____

This program is designed to print a formatted listing of all species codes and descriptions contained in the file "Species.New". This program gives the user the choice of outputting the entire file in ascending species code order or of selecting a major taxonomic group and outputting the list sorted by species description. This output is particularly useful for assigning codes to a list of species on field or contractor sheets. Examples of the two output formats are presented in Tables 3 and 4 respectively.

3.3 PRTCODES

This program is designed to print a formatted output of all method, parameter and special purpose codes. This output is sorted by ascending code. An example output from PRTCODES is presented in Table 5.

3.4 BLDSPEC

BLDSPEC is a specialized program which allows a user to search the file "Species.New" to determine whether or not a specific species or genus has been previously coded. If the genus or species in question is not found, BLDSPEC will allow the user to step through the program and build a new code. Upon completion of the code building sequence, the file "Species.New" is updated with the new code, description and reference. This program has been designed to overcome duplication of codes and has provisions which reduce the possibility of recording new species due to typographical errors.

Table 3

SPECIES DESCRIPTION FILE

SPECIES CODE	DESCRIPTION	REFERENCE
A——	PHYTOPLANKTON	
AA——	CHLOROPHYCEAE	
AAAA01	CHLAMYDOMONAS GLOBOSA	SNOW
AAAB01	GONIUM PECTORALE	MULLER
AAAC01	EUDORINA ELEGANS	EHRENBERG
AAAD01	GLOEOCYSTIS GIGAS	KUETZ. LAGERHEIM
AAAD02	GLOEOCYSTIS PLANKTONICA	WEST WEST LEMMERMANN
AAAE01	TETRASPORA GELATINOSA	VAUCHER DESVAUX
AAAE02	TETRASPORA LUBRICA	ROTH AGARDH
AAAE03	TETRASPORA GELATINOSA	VAUCHER DESVAUX
AAAF01	CHLOROCOCCUM HUMICOLA	NAEGELI RABENHORST
AAAG01	DICTYOSPHAERIUM EHRENBERGIANUM	NAEGELI
AAAG02	DICTYOSPHAERIUM PULCHELLUM	WOOD
AAAH01	PEDIASTRUM BORYANUM	TURPIN MENEGHINI
AAAH02	PEDIASTRUM DUPLEX VAR. CLATHRATUM	A. BRAUN LAGERHEIM
AAAH03	PEDIASTRUM TETRAS	EHRENBERG RALFS
AAAI01	COELASTRUM MICROSPORUM	NAEGELI
AAAI02	COELASTRUM PROBOSCIDEUM	BOHLIN
AAAI03	COELASTRUM CAMBRICA	ARCHER
AAAJ01	ANKISTRODESMUS FALCATUS VAR. ACICULARIS	WEST & WEST
AAAJ02	ANKISTRODESMUS SPIRALIS	TURNER LEMMERMANN
AAAK01	CHLORELLA ELLIPSOIDIA	GERNECK
AAAK02	CHLORELLA VULGARIS	BEYERINCK
AAAL01	OOCYSTIS BORGEI	SNOW
AAAL02	OOCYSTIS LACUSTRUS	CHODAT
AAAL03	OOCYSTIS SUBMARINA	LAGERHEIM
AAAM01	SELENASTRUM GRACILE	REINSCH
AAAN01	ACTINASTRUM HANTZCHII	LAGERHEIM
AAAO01	SCENEDESMUS BIJUGA	TURPIN LAGERHEIM
AAAO02	SCENEDESMUS DIMORPHUS	TURPIN KUETZING
AAAO03	SCENEDESMUS QUADRICAUDA	TURPIN DE BREBISSON
AAAO04	SCENEDESMUS BREVISPINA	GN SMITH CHODAT
AAAP01	TETRASTRUM STAUROGENIAEFORME	SCHRODER LEMM.
AAAQ01	GEMINELLA MINOR	NAEG. HEERING
AAAR01	HORMIDIUM KLEBSII	GM SMITH
AAAS01	ULOTHRIX AEQUALIS	KUETZING
AAAS02	ULOTHRIX ZONATA	WEBER&MOHR KUETZING
AAAS03	ULOTHRIX VARIABILIS	KUTZING
AAAT01	MICROSPORA STAGNORUM	KUETZ. LAGERHEIM
AAAT02	MICROSPORA WILLEANA	LAGERHEIM
AAAU01	MOUGEOTIA GENUFLEXA	DILLW. CA AGARDH
AAAU02	MOUGEOTIA SCALARIS	HASSALL
AAAU03	MOUGEOTIA VIRIDIS	KUETZING WITTROCK
AAAU04	MOUGEOTIA QUADRANGULATA	HASSALL
AAAV01	SPIROGYRA FLUVIATILIS	HILSE
AAAV02	SPIROGYRA WEBERI	KUETZING
AAAV03	SPIROGYRA VARIANS	HASSALL KUTZING
AAAW01	ZYGNEMA PECTINATUM	VAUCHER AGARDH
AAAX01	CLOSTERIDIUM ACUTUM	LYNGBYE BREBISSON

Table 4

SPECIES DESCRIPTION FILE

12-Aug-85 MAJOR TAXONOMIC GROUP - PISCES

SPECIES CODE	DESCRIPTION	REFERENCE
LBAA00	ACIPENSER BREVIROSTRUM	LESUEUR
LBAA01	ACIPENSER FULVESCENS	RAFINESQUE
LBAA02	ACIPENSER MEDIROSTRIS	AYRES
LBAA03	ACIPENSER OXYRHYNCHUS	MITCHILL
LBAA04	ACIPENSER TRANSMONTANUS	RICHARDSON
LB——	ACIPENSERIDAE (STURGEON)	
LXAA00	ACROCHEILUS ALUTACEUS	AGASSIZ & PICKERING
LEAA01	ALOSA PSEUDOHARENGUS	WILSON
LEAA02	ALOSA SAPIDISSIMA	WILSON
LUAA00	AMBLOPLITES RUPESTRIS	RAFINESQUE
LDAA00	AMIA CALVA	LINNAEUS
LD——	AMIIDAE (BOWFIN)	
LVAC00	AMMOCRYPTA PELLUCIDA	PUTNAM
LNAA00	ANGUILLA ROSTRATA	LESUEUR
LN——	ANGUILLIDAE (EELS)	
LEAA00	AOLSA AESTIVALIS	MITCHILL
LWAA00	APLODINOTUS GRUNNIENS	RAFINESQUE
LRAA00	APPELTES QUADRACUS	MITCHILL
LP——	ATHERINIDAE (SILVERSIDES)	
LXAB00	CARASSIUS AURATUS	LINNAEUS
LLAA00	CARPIODES CYPRINUS	LESUEUR
LL——	CATOSTOMIDAE (SUCKERS)	
LLAB00	CATOSTOMUS CATOSTOMUS	FORSTER
LLAB01	CATOSTOMUS COLUMBIANUS	EIGENMANN & EIGENMANN
LLAB02	CATOSTOMUS COMMERSONI	LACEPEDE
LLAB03	CATOSTOMUS MACROCHEILUS	GIRARD
LLAB04	CATOSTOMUS PLATYRHYNCHUS	COPE
LU——	CENTRARCHIDAE (SUNFISH)	
LXAC00	CHROSOMUS EOS	COPE
LXAC01	CHROSOMUS NEOGAEUS	COPE
LXAD00	CLINOSTOMUS ELONGATUS	KIRTLAND
LE——	CLUPEIDAE (HERRING)	
LFAD00	COREGONUS ALPENAE	KOELZ
LFAD01	COREGONUS ARTEDII	LESUEUR
LFAD02	COREGONUS AUTUMNALIS	PALLAS
LFAD13	COREGONUS CANADENSIS	SCOTT
LFAD11	COREGONUS CLUPEAFORMIS	MITCHILL
LFAD03	COREGONUS HOYI	GILL
LFAD04	COREGONUS JOHANNAE	WAGNER
LFAD05	COREGONUS KIYI	KOELZ
LFAD06	COREGONUS LAURETTAE	BEAN
LFAD12	COREGONUS NASUS	POLLAS
LFAD07	COREGONUS NIGRIPINNIS	GILL
LFAD08	COREGONUS REIGHARDI	KOELZ
LFAD09	COREGONUS SARDINELLA	VALENCIENNES
LFAD10	COREGONUS ZENITHICUS	JORDAN & EVERMANN
LX——	COTTIDAE (SCULPINS)	
LXAA00	COTTUS ALEUTICUS	GILBERT
LXAA01	COTTUS ASPER	RICHARDSON
LXAA02	COTTUS BAIRDI	GIRARD

Table 5

METHOD AND PARAMETER CODE DESCRIPTION FILE

18-Sep-85 PAGE 1

CODE	DESCRIPTION
MA00—	GENERAL-PHYTOPLANKTON
MA0000	BULK SAMPLE (ML)
MA0005	ACID LUGOLS PRESERVED (%)
MA0006	FORMALIN PRESERVED (%)
MA01—	NETS-PHYTOPLANKTON
MA0100	NET LENGTH (CM)
MA0101	NET MOUTH DIAMETER (CM)
MA0102	NET VOLUME (L)
MA0104	TOW LENGTH (M)
MA0105	TOW ORIENTATION
MA0106	CONICAL NET SAMPLER (MESH SIZE (U))
MC00—	GENERAL-ZOOPLANKTON
MC0000	BULK SAMPLE (ML)
MC0005	ACID LUGOLS PRESERVED (%)
MC0006	FORMALIN PRESERVED (%)
MC01—	NETS-ZOOPLANKTON
MC0101	NET LENGTH (CM)
MC0102	NET MOUTH DIAMETER (CM)
MC0104	TOW LENGTH (M)
MC0105	TOW ORIENTATION
MC0106	CONICAL NET SAMPLER (MESH SIZE (U))
MC0107	WISCONSIN NET (MESH SIZE U)
ML00—	GENERAL-FISH
ML0000	BULK SAMPLE (L)
ML0005	ACID LUGOLS PRESERVED (%)
ML0006	FORMALIN PRESERVED (%)
ML01—	NETS-FISH
ML0100	NET LENGTH (M)
ML0101	NET HEIGHT (M)
ML0102	NET MOUTH DIAMETER (M)
ML0103	NET VOLUME (L)
ML0104	MONOFILAMENT GILL NET (MESH SIZE (IN))
MMA0—	MERISTICS-PHYTOPLANKTON
MMA002	CONCENTRATION (CELLS/LITER)
MMA003	INDIVIDUALS /SAMPLE
MMC0—	MERISTICS-ZOOPLANKTON
MMC002	CONCENTRATION (CELLS/LITER)
MMC003	INDIVIDUALS/SAMPLE
MML0—	MERISTICS-FISH
MML003	INDIVIDUALS/SAMPLE
MML004	FORK LENGTH (CM)
MML005	TOTAL LENGTH (CM)
MML006	SEX (F OR M)
MML007	WEIGHT (KG)
MML008	AGE (YRS)
MMM0—	SAMPLING TIME INFORMATION
MMM000	SAMPLING DURATION (HRS)
MMM001	INITIATION DATE (D-M-Y)
MMM002	COMPLETION DATE (D-M-Y)
MMM1—	SAMPLING DEPTH INFORMATION

3.5 BIODAT

"BIODAT" is the main program in NABIODAT and has provisions for initializing, editing, viewing and reporting sample information for biological data. As was the case for the peripheral programs BIODAT is menu driven, having a main menu (Table 6) and a series of sub-menus.

The sample initialization procedure presents various screens which will provide user prompts and also indicate the related field size. The program requires the input of station description information and allows for the entry of as much biological data as is available (Table 7).

The "ADD" routine provides the ability to enter additional species, methods or parameter data to a sample that has been previously initialized. This is particularly useful for samples which are to be analyzed by contract as there may be considerable time delays between sampling and receipt of the results. The ADD routine uses the sample number as the primary key and provides a screen which enables rapid entry of new data (Table 8).

The editing capabilities of "BIODAT" are extremely powerful, allowing correction of all system information with the exception of the primary key (STATION #). The editor is comprised of two routines which enable the user to edit either sample data or description files. The description file editor is menu driven and has provisions for editing, adding or deleting a record from either the species code file or the method/parameter code file (Table 9). This procedure will enable the

correction of any typographical errors and the deletion of any duplicate codes. The sample data editor is designed to accommodate editing of either the sample station information or sample codes and values. This editor selects the sample using the primary key and allows correction of any record. In order to facilitate editing codes and values, the user can search the sample using one of three secondary keys (Table 10). In addition to the editing routine, there is also a delete program which will enable the user to delete a sample.

The final sub-routine of BIODAT is the "Report Sample" section. This allows the user to specify sample numbers and have an output report either listed on the screen or printed on a line printer. This routine will result in two forms of output depending on the type of sample data. Report style #1, lists station information and species codes with the corresponding values for a single parameter code (Table 11). Report style #2 lists station information, species codes and values for a series of parameter codes (Table 12). The output report style is chosen automatically and depends on the values entered for the variable "ID Number" in the initialization or add data routines. This ID number is a variable which enables the user to link a series of parameters and values to a specific individual in a sample. Thus samples with blank or "0" entered as ID number will be reported using Style 1 while samples with ID numbers greater than zero will be reported using Style 2.

Table 6

BIOLOGICAL DATA MANAGEMENT SYSTEM

MAIN MENU

1) Initialize Sample
2) ADD Data to a Sample
3) Report Sample
4) Edit Sample Data
5) Edit Description Files
6) Delete Sample
7) Exit Program

ENTER CHOICE ?

Table 7

Biological Data Management System

1.	Sample #	------
2.	Station #	----------
3.	Subid #	----
4.	Sampling Date "02-Jun-1983"	----------
5.	Sampling Time	----
6.	Time Zone	---
7.	Sample Description	----------
8.	Species Code	------
9.	ID Number	----
10.	Parameter Code	------
11.	Value	-------

Is the above information correct ?'(Y/N)'?

Do You Wish to Enter More Parameters - Values 'Y/N'?

Change Which Element? 'Type Element # '

Do You Wish to Initialize More Samples 'Y/N' ?

Table 8

Biological Data Management System

ADD Data to Sample

Enter Sample Number to Locate ? 100006

Sample # 100006 Station # 01NS01ED0005

Sample Description - Deep Station Sampled from Helicopter

Sampling Date - 17-Oct-1984 Time - 0950 AST

Sub I.D. # 0244

1. Species Code ? ------
2. I.D. Number ----
3. Parameter Code ------
4. Value -------

Is the Above Information Correct ? 'Y/N' ?

Do You Wish to Enter More Data for This Sample 'Y/N' ?

Do You wish to Enter Data for Another Sample 'Y/N' ?

Table 9

CODE DESCRIPTION FILE EDITOR

1) Edit SPECIES.NEW

2) Edit BIOCODE.NEW

3) Add a Record to SPECIES.NEW

4) Add a Record to BIOCODE.NEW

5) Delete a Record from SPECIES.NEW

6) Delete a Record from BIOCODE.NEW

7) Return to Main Menu

CHOICE ?

1) Enter 6 Character Species Code ?

SPECIES CODE	DESCRIPTION
AAAA01	CHLAMYDOMONAS GL0B0SA

REFERENCE	SNOW

1) Description 2) Reference 3) Exit

CHOICE ?

Do You Wish to Edit More Codes ?

3) Enter 6 Character Species Code ?

Type Description (Max 40 Characters) - ?
Type Reference (Max 40 Characters) - ?

THIS RECORD HAS BEEN ADDED TO SPECIES.NEW

Do You Wish to Add More Species Codes ? (Y/N) ?

5) Type Species Code to be Deleted ?

Species Code '_____ ' has been deleted

Do You Wish to Delete More Species Codes ? (Y/N) ?

Table 10

SAMPLE EDITING MENU

1. Edit Sampling Information
2. Edit Codes or Values
3. Return to Main Menu

1.

SAMPLE #	STATION #	SUBID	SDATE	STIME	T ZONE
100006	01NS01ED0005	0244	17-Oct-1984	0950	AST

SAMDES	DEEP STATION SAMPLED FROM HELICOPTER

1) STATION # 2) SUBID 3) SDATE 4) STIME
5) T ZONE 6) SAMDES 7) ANOTHER SAMPLE 5) EXIT

CHOICE: ?

ENTER NEW ?

2. Enter sample # to Locate ?

Key ON -- 1) Species Code
2) Parameter Code
3) ID Number

Enter 6 Character Parameter Code

SPCODE	IDNUM	PARCODE	VALUE
AAAA01	0	MA0107	10

1) SPCODE 2) IDNUM 3) PARCODE 4) VALUE
5) ANOTHER SAMPLE 6) NEXT MATCH
7) SELECT ANOTHER CODE 8) DELETE RECORD 9) EXIT

CHOICE: ?

ENTER NEW

Table II

SAMPLE #: 100003 PAGE 1

SAMPLE DESCRIPTION : DEEP STATION SAMPLED FROM HELICOPTER

SAMPLING DATE : 17-OCT-1984 TIME : 1000 ZONE : AST

STATION # : 01NS01ED0002

DESCRIPTION : BIG DAM WEST LAKE IN MIDDLE

SUBID : 0244

CODE	DESCRIPTION	VALUE
CFAD01	GASTROPUS STYLIFER	40
CFBI--	SYNCHAETA SP.	40
CFAG11	KERATELLA TAUROCEPHALA	3760
CFAG00	KERATELLA COCHLEARIS	1000
CFAF01	KELLICOTTIA BOSTONIENSIS	280
CFCD01	CONOCHILUS UNICORNIS	360
CFCP--	CONOCHILOIDES SP.	40
CFAK04	TRICHOCERCA MULTICRINIS	40
CFAA00	POLYARTHRA VULGARIS	120
CFAA05	POLYARTHRA DOLICHOPTERA	80
CFAA01	POLYARTHRA EURYPTERA	240
CFDS00	PLOESOMA LENTICULARE	40
CEAP00	BOSMINA LONGIROSTRIS	480
CEBT00	EUBOSMINA LONGISPINA	300
CEBT01	EUBOSMINA TUBICEN	10
CEAH00	HOLOPEDIUM GIBBERUM	155
CEAE00	DIAPHANOSOMA BRACHYURUM	5
CAAG48	DIAPTOMUS MINUTUS	15
CAAG47	DIAPTOMUS FLORIDANUS	160
CBAE01	TROPOCYCLOPS PRASINUS MEXICANUS	40
CAA---	CALANOID COPEPODIDS	1520
CAB---	CALANOID NAUPLII	5
CBA---	CYCLOPOID COPEPODIDS	90
CBB---	CYCLOPOID NAUPLII	15
MMC003	INDIVIDUALS/SAMPLE	
MC0101	NET LENGTH (CM)	75
MC0104	TOW LENGTH (M)	3.5
MC0105	TOW ORIENTATION	VERT
MC0107	WISCONSIN NET (MESH SIZE U)	63
MC0102	NET MOUTH DIAMETER (CM)	20
MMM101	SAMPLE DEPTH RANGE (Z1-Z2 (M))	0-3.5
MMMMMM	NAQUADAT DATA SAMPLE #	402701
MMMMMB	ADDITIONAL BIOLOGICAL DATA (SAMPLE #)	100CC4

REMARKS

Table 12

SAMPLE #: 100000 PAGE 1

SAMPLE DESCRIPTION : FISH COLLECTIONS AT FOUR STATIONS

SAMPLING DATE : 14-SEP-1983 TIME : 1200 ZONE : AST

STATION # : 01NF02XB0002

ESCRIPTION : ST.AUGUSTIN R. HEADWATER L.

SUBID : 0085

ID	CODE	DESCRIPTION	VALUE
1	LJAA02	ESOX LUCIUS	
1	MML007	WEIGHT (KG)	7.5
1	MML004	FORK LENGTH (CM)	81.9
1	MML005	TOTAL LENGTH (CM)	86.4
1	MML006	SEX (F OR M)	F
2	LJAA02	ESOX LUCIUS	
2	MML007	WEIGHT (KG)	6.5
2	MML004	FORK LENGTH (CM)	74.3
2	MML005	TOTAL LENGTH (CM)	79.7
2	MML006	SEX (F OR M)	M
3	LJAA02	ESOX LUCIUS	
3	MML007	WEIGHT (KG)	6.5
3	MML004	FORK LENGTH (CM)	76.2
3	MML005	TOTAL LENGTH (CM)	81.3
3	MML006	SEX (F OR M)	M
4	LJAA02	ESOX LUCIUS	
4	MML007	WEIGHT (KG)	5.8
4	MML004	FORK LENGTH (CM)	67.3
4	MML005	TOTAL LENGTH (CM)	74.9
4	MML006	SEX (F OR M)	M
5	LJAA02	ESOX LUCIUS	
5	MML007	WEIGHT (KG)	5.5
5	MML004	FORK LENGTH (CM)	76.8
5	MML005	TOTAL LENGTH (CM)	80.
5	MML006	SEX (F OR M)	F
6	LJAA02	ESOX LUCIUS	
6	MML007	WEIGHT (KG)	13.
6	MML004	FORK LENGTH (CM)	94.
6	MML005	TOTAL LENGTH (CM)	101.6
6	MML006	SEX (F OR M)	F
	ML0104	MONOFILAMENT GILL NET (MESH SIZE (IN))	.5
	ML0104	MONOFILAMENT GILL NET (MESH SIZE (IN))	1
	ML0100	NET LENGTH (M)	90
	ML0101	NET WIDTH (M)	8
	ML0000	SAMPLING DURATION (HRS)	24

4.0 System Integrity

NABIODAT has been designed as a self contained system specifically for the use of the Water Quality Branch, Atlantic Region. In spite of this, the system will be made available to other agencies and in the future may form the basis of a National biological data management system. As the system develops and the number of users increases, it will be necessary to restrict access to the code building routines. Failure to control the designation of new species, method and parameter codes will lead to code duplication which will severely limit the interpretation of regional, national and multi-agency biological data. As a first step at controlling code designation, the program listing for the BLDSPEC program will not be presented in this report. At present, arrangements can be made with the authors to permit access to BLDSPEC or to have new codes established.

As the system demand increases, it is felt that the code designation and development should be controlled by a central agency in Ottawa. The Water Quality Branch in Ottawa presently has experience in the management of NAQUADAT and thus would be the logical co-ordinators of the NABIODAT system. With this co-ordination it is felt the NABIODAT can become a flexible, large scale biological data management system.

5.0 Program Listing

This section presents the program listing for NABIODAT, the biological data management system. These listing include not only BIODAT, the main program, but also presents three of the four peripheral programs.

```
5 REM ***************************************************************************
    ! *                    N A B I O D A T                                      *
    ! *                                                                         *
    ! * B I O L O G I C A L   D A T A   M A N A G E M E N T   S Y S T E M       *
    ! *                                                                         *
    ! *                                                                         *
    ! *      DESIGNED BY : MONITORING AND SURVEYS DIVISION *                    *
    ! *                                                                         *
    ! *      PROGRAMMED BY : DATA SYSTEMS SECTION **                            *
    ! *                                                                         *
    ! *                     WATER QUALITY BRANCH                                *
    ! *                        MONCTON , N.B.                                   *
    ! *                                                                         *
    ! *                                                                         *
    ! *      CONTRIBUTORS :   G.D.HOWELL    *                                   *
    ! *                       L.D.WONG      **                                  *
    ! *                       J.R.FENTON    **                                  *
    ! *                       R.A.BINGHAM   **                                  *
    ! *                                                                         *
    ! *                                                                         *
    ! *  THIS SYSTEM HAS PROVISIONS FOR INITIALIZING ,EDITING ,VIEWING,         *
    ! * AND REPORTING SAMPLE INFORMATION FOR BIOLOGICAL DATA.                   *
    ! *  THIS PROGRAM IS ONE OF A NUMBER OF PROGRAMS WHICH MAKEUP THE           *
    ! * BIOLOGICAL DATA MANAGEMENT SYSTEM. FOR SAMPLE INITIALIZATION            *
    ! * THE VARIOUS SCREENS WILL PROVIDE PROMPTS ,AND ALSO INDICATE THE         *
    ! * RELATED FIELD SIZE . VARIOUS MENUS ARE AVAILABLE TO HELP THE            *
    ! * USER WITH THE EDITING,VIEWING,AND REPORTING FEATURES OF THIS            *
    ! * SYSTEM.                                                                 *
    ! *                                                                         *
    ! *  OTHER PROGRAMS INCLUDED IN THE BIOLOGICAL DATA MANAGEMENT              *
    ! * SYSTEM ARE :                                                            *
    ! *                                                                         *
    ! *      1) BIOSHEET.BAS      CREATES FIELD AND CONTRACTOR SHEETS           *
    ! *      2) BLDSPEC.BAS       BUILDS NEW SPECIES CODES                      *
    ! *      3) PRTSPEC.BAS       PRODUCES A FORMATTED LISTING OF               *
    ! *                             'SPECIES.NEW'                               *
    ! *      4) PRTCODES.BAS      PRODUCES A FORMATTED LISTING OF               *
    ! *                             'BIOCODE.NEW'                               *
    ! *                                                                         *
    ! ***************************************************************************
    !
    !
10 ONERROR GOTO 30000
11 !    ***************************************************
   !    * INITIALIZE VARIABLES AND SET ESCAPE SEQUENCES *
   !    ***************************************************
15 DECLARE STRING MOV,LFEED,CLRSCREEN,PROMPT,PROMPT1,PROMPT2,PROMPT3,PROMPT4,&
       PROMPT5,PROMPT6,PROMPT7,PROMPT8,PROMPT9,PROMPT10,TITLE,SAM1,STATION1,&
       SUBID1,SDATE1,STIME1,TZONE1,SAMDES1,SPCODE1,PARCODE1,PVAL1,RES,ERASE,&
       MONTH(12),IDNUM1,BELL,L1,L2,L3,L4,L5,L6,SDAY,SMON,SYR,INCODE,RES1,&
       MESS1,MESS2,ERASE1,DESC1,REFER2
20 DECLARE INTEGER SAMIN,CHOICE,CFLAG,CHFLAG
22 DECLARE STRING SORT_TABLE(5000,5)
   SORT_TABLE(0,4) = "0"
   SORT_TABLE(0,5) = "^"
24 ROWW = 1
30 MOV=CHR$(27)+"["
       LFEED=CHR$(10)
       CLRSCREEN=CHR$(27)+"[H"+CHR$(27)+"[2J"
       ERASE=CHR$(27)+"[0J"
       BELL=CHR$(7)
       ERASE1 = CHR$(27) + "K"
40 FOR I=1 TO 12
```

```
        READ A$
        MONTH(I)=A$
      NEXT I
41 DATA "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"
50 TITLE="B I O L O G I C A L   D A T A   M A N A G E M E N T   S Y S T E M"
55 PROMPT=" 1) SAMPLE #                                    _____"
        PROMPT1=" 2) STATION #                              __________"
        PROMPT2=" 3) SUBID #                                ___"
        PROMPT3=" 4) SAMPLING DATE '02-JUN-1985'            __________"
        PROMPT4=" 5) SAMPLING TIME                          ____"
        PROMPT5=" 6) TIME ZONE                              ___"
        PROMPT6=" 7) SAMPLE DESCRIPTION                     ____________________________________________"
        PROMPT7=" 8) SPECIES CODE                           ______"
        PROMPT8=" 9) ID NUMBER                              ____"
        PROMPT9="10) PARAMETER CODE                         ______"
        PROMPT10="11) VALUE                                  _______"
60 MESS1="      ******** CODE NOT ON FILE ********"
        MESS2="    ******** DUPLICATE KEY DETECTED **********"
65 MAP (RECTHR) STRING SCODE=6%,DES=4%,DES1=36%,REFER=40%
        MAP(RECTHR) STRING SCODE=6%,DESCRIP=40%,REFER=40%
        MAP (RECFOR) STRING BICODE=6%,DESCR=50%
69 !   ***********************************************
   !   * MAP AND OPEN THE OUTPUT FILE 'BIODAT.NEW *
   !   ***********************************************
70 MAP (RECONE) INTEGER SAMPLE,STRING STATION=12%,SUBID=4%,SDATE=11%,&
        STIME=4%,TZONE=3%,SAMDES=40%,SPCODE=6%,IDNUM=4%,PARCODE=6%,PVAL=7%
72 OPEN "BIODAT.NEW" AS FILE #1% ,&
        ORGANIZATION INDEXED FIXED ,&
        PRIMARY KEY SAMPLE DUPLICATES ,&
        ALTERNATE KEY SPCODE DUPLICATES CHANGES ,&
        ALTERNATE KEY IDNUM DUPLICATES CHANGES,&
        ALTERNATE KEY PARCODE DUPLICATES CHANGES ,&
        ACCESS MODIFY,MAP RECONE,ALLOW MODIFY
75 MAP (RECTWO) STRING WQSTN=12%,WRSTN=7%,STNDES=64%
        OPEN "[LABS11]STNFIL.NEW"AS FILE #2%,&
            ORGANIZATION INDEXED FIXED,&
            PRIMARY KEY WQSTN,&
            ALLOW MODIFY,MAP RECTWO,ACCESS MODIFY
     ! 80 OPEN "TXCO:" FOR OUTPUT AS FILE #9
      !
100 MAIN_MENU:
    REM *******************************************************
      ! *          CLEAR SCREEN AND DISPLAY MAIN MENU         *
      ! *******************************************************
105 PRINT CLRSCREEN;LFEED;LFEED
106     PRINT TAB(8);TITLE
        PRINT LFEED;LFEED
        PRINT TAB(36);"MAIN MENU"
        PRINT LFEED
        PRINT TAB(30);"1)  INITIALIZE SAMPLE"
        PRINT TAB(30);"2)  ADD DATA TO A SAMPLE "
        PRINT TAB(30);"3)  REPORT SAMPLE "
        PRINT TAB(30);"4)  EDIT SAMPLE DATA "
        PRINT TAB(30);"5)  EDIT DESCRIPTION FILES"
        PRINT TAB(30);"6)  DELETE SAMPLE "
        PRINT TAB(30);"7)  EXIT PROGRAM"
        PRINT LFEED
110 INPUT "    ENTER CHOICE - ";CHOICE
115 ON CHOICE GOSUB INIT,ADD_DATA,REPORT_SAM,EDIT_SAM,EDIT_DES,DEL_SAM,LEAVE
        GOTO MAIN_MENU
118 !   ********************************************
    !   * CREATE SCREEN FOR INITIALIZING SAMPLES *
    !   ********************************************
```

```
    !
119 INIT:
        PRINT CLRSCREEN
120 PRINT TAB(8);TITLE
        PRINT LFEED
        PRINT TAB(30);"INITIALIZATION MODE"
        PRINT LFEED
        PRINT TAB(6);PROMPT
        PRINT TAB(6);PROMPT1
        PRINT TAB(6);PROMPT2
        PRINT TAB(6);PROMPT3
        PRINT TAB(6);PROMPT4
        PRINT  TAB(6);PROMPT5
        PRINT  TAB(6);PROMPT6
        PRINT  TAB(6);PROMPT7
        PRINT  TAB(6);PROMPT8
        PRINT  TAB(6);PROMPT9
        PRINT  TAB(6);PROMPT10
125 CFLAG=0 \ CHFLAG = 0
129 !   ******************************************
    !   * CREATE INPUT SCREEN FOR INITIALIZATION *
    !   ******************************************
130 PRINT MOV+"7;35H"
        INPUT"       1) SAMPLE #                    ";SAM1
        IF SAM1="" THEN 140
        END IF
        IF LEN(SAM1)()6 THEN
          PRINT BELL;BELL
          GOTO 130
        END IF
        SAMPLE=VAL%(SAM1)
        GET #1, KEY #0 EQ SAMPLE
        PRINT BELL;BELL
        GOTO 130
135     IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
140 PRINT MOV+"8;35H"
        INPUT"       2) STATION #                   ";STATION1
        IF STATION1="" THEN 150
        END IF
        FIND #2%,KEY #0 EQ STATION1
        STATION=STATION1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
150 PRINT MOV+"9;35H"
        INPUT"       3) SUBID #                     ";SUBID1
        IF SUBID1="" THEN 160
        END IF
        IF LEN(SUBID1)()4 THEN
          PRINT BELL;BELL
          GOTO 150
        END IF
        SUBID=SUBID1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
160 PRINT MOV+"10;35H"
        INPUT"       4) SAMPLING DATE '02-JUN-1985'  ";SDATE1
        IF SDATE1="" THEN 170
        END IF
        IF LEN(SDATE1)()11 THEN
```

```
          PRINT BELL;BELL
          GOTO 160
        END IF
        SDATE=SDATE1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
170 PRINT MOV+"11;35H"
        INPUT"       5) SAMPLING TIME                ";STIME1
        IF STIME1="" THEN 180
        END IF
        IF LEN(STIME1)()4 THEN
          PRINT BELL;BELL
          GOTO 170
        END IF
        STIME=STIME1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
180 PRINT MOV+"12;35H"
        INPUT"       6) TIME ZONE                    ";TZONE1
        IF TZONE1="" THEN 190
        END IF
        IF LEN(TZONE1)()3 THEN
          PRINT BELL;BELL
          GOTO 180
        END IF
        TZONE=TZONE1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
190 PRINT MOV+"13;35H"
        INPUT"       7) SAMPLE DESCRIPTION           ";SAMDES1
        IF SAMDES1="" THEN 200
        END IF
        IF LEN(SAMDES1))40 THEN
          PRINT BELL;BELL
          PRINT MOV+"20;30H";"DESCRIPTION TOO LONG"
          SLEEP 3
          PRINT MOV+"20;30H";ERASE
          GOTO 190
        END IF
        SAMDES=SAMDES1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
200 PRINT MOV+"14;35H"
        INPUT"       8) SPECIES CODE                 ";SPCODE1
        IF SPCODE1="" THEN 210
        END IF
        IF LEN(SPCODE1)()6 THEN
          PRINT BELL;BELL
          GOTO 200
        END IF
        SPCODE=SPCODE1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
210 PRINT MOV+"15;35H"
        INPUT"       9) ID NUMBER                    ";IDNUM1
        IF IDNUM1="" THEN IDNUM1="0000"
        GOTO 220
        END IF
```

```
        IF LEN(IDNUM1))4 THEN
          PRINT BELL;BELL
          GOTO 210
        END IF
        IDNUM=IDNUM1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
220 PRINT MOV+"16;35H"
        INPUT"      10) PARAMETER CODE                ";PARCODE1
        IF PARCODE1="" THEN 230
        END IF
        IF LEN(PARCODE1)()6 THEN
          PRINT BELL;BELL
          GOTO 220
        END IF
        PARCODE=PARCODE1
        IF CFLAG=1 THEN PRINT MOV+"20;30H"
        PRINT ERASE
        GOTO 250
230 PRINT MOV+"17;35H"
        INPUT"      11) VALUE                         ";PVAL1
        IF PVAL1="" THEN 248
        END IF
        IF LEN(PVAL))7 THEN
          PRINT BELL;BELL
          GOTO 230
        END IF
        PVAL=PVAL1
248 PRINT LFEED
249 !   ********************************************************
    !   * CHECK IF INFORMATION ENTERED ON SCREEN IS CORRECT.  *
    !   * IF NOT ALL CORRECT THEN SELECT ELEMENT TO CHANGE.   *
    !   * REPEAT UNTIL ALL DATA ARE CORRECT.                  *
    !   *  IN ORDER TO CHANGE THE VALUE OF A FIELD YOU MUST   *
    !   * RE-TYPE THE COMPLETE FIELD.                         *
    !   ********************************************************
250 INPUT " IS THE ABOVE INFORMATION CORRECT ? (Y/N)";RES
255 IF RES="Y" THEN 280 \ END IF
260 INPUT" CHANGE WHICH ELEMENT ? 'TYPE ELEMENT #'";ELNUM
       CFLAG=1
265 PRINT MOV+"19;35H";ERASE
270 ON ELNUM GOTO 130,140,150,160,170,180,190,200,210,220,230
280 PUT #1                      !======) WRITE RECORD TO FILE (======
290 PRINT MOV+"19;35H";ERASE
295 INPUT "DO YOU WISH TO ENTER MORE PARAMETERS-VALUES 'Y/N' ";RES
       IF RES="N" THEN 300
      ELSE
       CFLAG = 0
       GOTO 200
300 PRINT LFEED
      INPUT "DO YOU WISH TO INITIALIZE MORE SAMPLES 'Y/N' ";RES
       IF RES="Y" THEN 119
      ELSE
       GOTO MAIN_MENU
      END IF
499 !  ************************************
    !  * ROUTINE TO ADD DATA TO A SAMPLE *
    !  ************************************
    !
    !        ************** END INIT ***************
    !
500 ADD_DATA:
```

```
501 REM ***********************************************
    ! * CREATE SCREEN FOR ADDING DATA TO SAMPLE *
    ! ***********************************************
510 PRINT CLRSCREEN
      PRINT TAB(8);TITLE;LFEED
      PRINT TAB(31);"ADD DATA TO SAMPLE";LFEED
520 INPUT "ENTER SAMPLE NUMBER TO LOCATE ";SAMIN
      PRINT MOV+"5;30H"
      PRINT ERASE
525 GET #1%,KEY #0 EQ SAMIN
530 PRINT MOV+"6;5H";"SAMPLE # - ";SAMPLE;"                              STATION # - ";&
      STATION ;LFEED
540 PRINT MOV+"8;5H";"SAMPLE DESCRIPTION - ";SAMDES;LFEED
550 PRINT MOV+"10;5H";"SAMPLING DATE - ";SDATE;"                  TIME - ";&
      STIME;"  ";TZONE ;LFEED
      PRINT MOV+"12;5H";"SUBID # ";SUBID
555 PRINT LFEED
      PRINT TAB(10);"*********************************************************"
560 PRINT LFEED
570 PRINT TAB(6);" 1) ";SEG$(PROMPT7,5,LEN(PROMPT7))
      PRINT TAB(6);" 2) ";SEG$(PROMPT8,5,LEN(PROMPT8))
      PRINT TAB(6);" 3) ";SEG$(PROMPT9,5,LEN(PROMPT9))
      PRINT TAB(6);" 4) ";SEG$(PROMPT10,5,LEN(PROMPT10))
      CFLAG=0
579 REM ************************************
    ! * CREATE INPUT PORTION OF SCREEN *
    ! ************************************
580 PRINT MOV+"17;30H"
      INPUT"       1) SPECIES CODE                ";SPCODE1
      IF SPCODE1="" THEN 600
      END IF
      IF LEN(SPCODE1)()6 THEN PRINT BELL
       GOTO 580 \ END IF
      SPCODE=SPCODE1
      IF CFLAG=1 THEN 700
600 PRINT MOV+"18;30H"
      INPUT"       2) ID NUMBER                   ";IDNUM1
      IF IDNUM1="" THEN 620
      END IF
      IF LEN(IDNUM1))4 THEN PRINT BELL
       GOTO 600 \ END IF
      IDNUM=IDNUM1
      IF CFLAG=1 THEN 700
620 PRINT MOV+"19;30H"
      INPUT"       3) PARAMETER CODE              ";PARCODE1
      IF PARCODE1 = "" THEN 640
      END IF
      IF LEN(PARCODE1)()6 THEN PRINT BELL
       GOTO 620 \ END IF
      PARCODE=PARCODE1
      IF CFLAG=1 THEN 700
640 PRINT MOV+"20;30H"
      INPUT"       4) VALUE                       ";PVAL1
      IF PVAL1 = "" THEN 699
      END IF
      IF LEN(PVAL1)) 7 THEN PRINT BELL
       GOTO 640 \ END IF
      PVAL=PVAL1
699 REM *****************************************************
    ! * MOVE CURSOR TO LOWER PORTION OF SCREEN AND ERASE *
    ! * AFTER DISPLAYING EACH QUESTION.                  *
    ! *****************************************************
700 PRINT MOV+"22;30H";ERASE
```

```
        INPUT "   IS THE ABOVE INFORMATION CORRECT ? 'Y/N' ";RES
        IF RES="N" THEN PRINT MOV+"22;30H";ERASE
         INPUT "   CHANGE WHICH ELEMENT  ";ELNUM
         CFLAG=1
         ON ELNUM GOTO 580,600,620,640
        END IF
    IF IDNUM1 = "" THEN
       IDNUM = "0000"
    END IF
    IF SPCODE = "" THEN
       GOTO 500
    END IF
720 PUT #1                              !====) WRITE NEW RECORD TO FILE (=====
730 PRINT MOV+"22;30H";ERASE
        INPUT"   DO YOU WISH TO ENTER MORE DATA FOR THIS SAMPLE 'Y/N' ";RES
        IF RES="Y" THEN
         PRINT MOV+"17;30H"
           CFLAG = 0
        GOTO 580
740 PRINT MOV+"22;30H";ERASE
        INPUT"   DO YOU WISH TO ENTER DATA FOR ANOTHER SAMPLE 'Y/N' ";RES
        IF RES="Y" THEN
           CFLAG = 0
           GOTO 510
        ELSE
        GOTO 105
       !            ************** END ADD_DATA *************
       !
1000 REPORT_SAM:
1005    REM ** * * * * * ************************************ * * * * * **
           *     THIS ROUTINE IS DESIGNED TO PRODUCE A GROUP OF       *
           * REPORTS. THE USER IS ASKED FOR THE SAMPLE NUMBERS OF THE *
           * RECORDS OF WHICH A REPORT IS WANTED AND PRODUCE A REPORT *
           * OF THE TYPE DESIGNED BY THE WAY THE DATA IS ARRANGED IN  *
           * THE FILE CONTAINING THE SAMPLE NUMBER .                  *
           ** * * * * * ************************************ * * * * * **
1020    DECLARE STRING CLRPART, SAM_ARRAY(50),  &
                       METHOD_VECTOR(5000,3), GET_IDNUM, GET_SPEC, DOT_LINE
        DECLARE INTEGER I _POS, GET_SAM, SAM, POSITION
1050    CLRPART = MOV + "0J"
        DOT_LINE = "-------------------------------------------------------------------------------------------------------------"
1100    REM ** * * * * * ************************************ * * * * * **
           *   SET UP THE SCREEN AND PERFORM THE ENTERING OF SAMPLE   *
           *  NUMBERS.                                                *
           ** * * * * * ************************************ * * * * * **
1110    PRINT CLRSCREEN,MOV + "10H"
        I = 1
        PRINT TAB(10);"B I O L O G I C A L   R E P O R T  M E N U E "
        PRINT
        PRINT
        PRINT TAB(10);"REMINDER:     TO EXIT FROM ANY QUESTION ENTER '$'"
        PRINT
1120    PRINT MOV + "18H",,,,,,,ERASE1;MOV + "17H"
        INPUT "ENTER ALL OR PART OF THE SAMPLE NUMBER ( FOR REPORT ) ";SAM$
        IF SAM$ = "$" THEN
           GOTO 1230
        END IF
        IF LEN(SAM$) ( 6 THEN
           IF I () 1 THEN
            SAM$ = SEG$(SAM_ARRAY(I-1),1,LEN(SAM_ARRAY(I-1)) - LEN(SAM$)) + SAM$
           END IF
        END IF
1186    SAM = VAL%(SAM$)
```

```
1190   GET #1, KEY #0 EQ SAM
1200   SAM_ARRAY(I) = SAM$
       PRINT MOV + "21H",CLRPART
       PRINT " LAST VALID SAMPLE NUMBER WAS ";SAM_ARRAY(I)
       I = I + 1
       GOTO 1120
1230   REM ** * * * * * ************************************ * * * * * **
           *            ASK FOR THE NUMBER OF COPIES WANTED          *
           ** * * * * * ************************************ * * * * * **
1231   PRINT MOV + "17H",CLRPART
       INPUT "         DO YOU WANT A HARD COPY REPORT FOR ALL THE SAMPLES ";ANS$
1233    SELECT ANS$
         CASE "Y","YES"
           OPEN "TXCO:" FOR OUTPUT AS FILE #8
           PRINT #8, CHR$(27) + '?" '
           PRINT #8, CHR$(12)
           PRINT_KEY = 1
         CASE "N","NO"
           MAP (PREC) STRING MESSAGE=132%
           KILL "PRTFILE.DAT"
           OPEN "PRTFILE.DAT" AS FILE #8, &
             ORGANIZATION SEQUENTIAL VARIABLE, &
             RECORDSIZE 132%, &
             MAP PREC, ACCESS MODIFY
           PRINT_KEY = 0
         CASE ELSE
           GOTO 1231
       END SELECT
1240   PRINT MOV + "17H",CLRPART
1250   INPUT "         HOW MANY COPIES WOULD YOU LIKE ";ANS$
1260   ANS = VAL%(ANS$)
1270   IF ANS ( 0 THEN
          GOTO 1240
       END IF
1300   REM ** * * * * * ************************************ * * * * * **
           *              DETERMINE THE REPORT TYPE                  *
           *   READ THROUGH THE FILE LOOKING FOR AN  ID NUMBER       *
           *   IF ID NUMBER IS FOUND THEN THE REPORT FORM IS         *
           *            HEADING === COMMON IN BOTH REPORTS           *
           *            I CODE DESCRIPTION VALUE                     *
           *            1 KLK'  LKKL;       2                        *
           *            - ---  ---------  --                        *
           *           ----------------------                        *
           *   ELSE                                                  *
           *            ID  CODE  DESCRIPTION   VALUE                *
           *              __   _______      ___                      *
           ** * * * * * ************************************ * * * * * **
1305   MAP (SPECIES) STRING SPCODES=6%,SPEC=4%,SPEC1=36%,REFER1=40%
        MAP(SPECIES) STRING SPCODES=6%,SPEC2=40%,REFER1=40%
1307   OPEN "SPECIES.NEW" AS FILE #5, &
         ORGANIZATION INDEXED FIXED, &
         PRIMARY KEY SPCODES, &
         ALTERNATE KEY SPEC DUPLICATES, &
         ALLOW MODIFY, MAP SPECIES, ACCESS MODIFY
1309   OPEN "BIOCODE.NEW" AS FILE #6, &
         ORGANIZATION INDEXED FIXED, &
         PRIMARY KEY BICODE, &
         ALLOW MODIFY, MAP RECFOR, ACCESS MODIFY
1310   FOR K = 1 TO ANS
         FOR G = 1 TO I - 1
           GET_SAM = VAL%(SAM_ARRAY(G))
1320       GET #1, KEY #0 EQ GET_SAM
           GOSUB PRINT_HEADER
```

```
        POSITION = 0
1330    REM ***************************************************
            * READ ALLRECORDS FOR A SAMPLE NUMBER CHECK FOR *
            * AN ID NUMBER OTHER THEN 0                     *
            ***************************************************
1340    WHILE GET_SAM = SAMPLE
1350      IF VAL%(IDNUM) ) 0 THEN
            POSITION = 1
          END IF
1360      GET #1
1370    NEXT
1375    RESET #1
        GET #1, KEY #0 EQ GET_SAM
        M_POS = 1
1380    IF POSITION = 0 THEN
           PRINT #8; TAB(6);"CODE";TAB(31);"DESCRIPTION";TAB(73);"VALUE"
           PRINT #8; TAB(6);"----";TAB(31);"-----------";TAB(73);"-----"
           GOSUB REPORT_TYPE_1
        ELSE
           PRINT #8, TAB(6);"ID";TAB(14);"CODE";TAB(36);"DESCRIPTION";TAB(73);"VALUE"
           PRINT #8, TAB(6);"--";TAB(14);"----";TAB(36);"-----------";TAB(73);"-----"
           GOSUB REPORT_TYPE_2
        END IF
        GOSUB PRINT_REMARKS
1390    RESET #1
1400    NEXT G
        PRINT #8,CHR$(12)
1410  NEXT K
      !*********************************************************
      !*       CLOSE THE PRINT AND GO BACK TO THE MENU       *
      !*********************************************************
1411  IF PRINT_KEY = 1 THEN
         CLOSE #8
         GOTO 1490
      END IF
1416  !*********************************************************
      !*      PRINT WHAT EVER IS IN THE "PRTFILE.DAT" TO THE *
      !*                    SCREEN                           *
      !*********************************************************
1420  RESET #8
      PRINT CLRSCREEN
1430  GET #8
      IF MESSAGE = CHR$(12) THEN
         SLEEP 1%
         PRINT CLRSCREEN
      END IF
      PRINT MESSAGE
      GOTO 1430
1490  CLOSE #8
      GOTO MAIN_MENU
1500  REM ** * * * * * ********************************** * * * * * **
          *                    PRINT_HEADER                        *
          *      THIS ROUTINE PRINT THE STATION AND SAMPLE DATA AT  *
          *  THE TOP THE FIRST PAGE.                               *
          ** * * * * * ********************************** * * * * * **
1510  PRINT_HEADER:
1515    GET #2,KEY #0 EQ STATION
1520    TITLE$ = "B I O L O G I C A L   R E P O R T   L I S T I N G"
        PRINT #8, TAB(16);TITLE$
        PRINT #8 \ PRINT #8
        PAGE$ = "1"
1530    PRINT #8, TAB(6);"SAMPLE #: ";SAMPLE;TAB(65);"PAGE ";PAGE$
        PRINT #8
```

```
1540      PRINT #8, TAB(10);"SAMPLE DESCRIPTION : ";SAMDES
          PRINT #8
1545      PRINT #8, TAB(10);"SAMPLING DATE : ";SDATE;TAB(43);"TIME : ";STIME;TAB(60);"ZONE : ";TZONE
          PRINT #8
1550      PRINT #8, TAB(10);"STATION # : ";STATION
          PRINT #8
1560      PRINT #8, TAB(1);"DESCRIPTION : ";STNDES
          PRINT #8
1580      PRINT #8, TAB(34);"SUBID : ";SUBID
1590      PRINT #8
1600      PRINT #8, DOT_LINE
1610      PRINT #8
          LINE_COUNT = 17
1620    PRINT_RETURN:
          RETURN
1700    REM ** * * * * * ************************************ * * * * * **
            *                        PRINT_REMARKS                       *
            ** * * * * * ************************************ * * * * * **
1710    PRINT_REMARKS:
            PRINT #8 \ PRINT #8
            PRINT #8," REMARKS  ----------------------------------------------------------------------
            PRINT #8
            PRINT #8, TAB(1);DOT_LINE
            PRINT #8
            PRINT #8, TAB(1);DOT_LINE
            PRINT #8, CHR$(12)
        REMARKS_RETURN:
            RETURN
1800    REM ** * * * * * ************************************ * * * * * **
            *                        MAY_NEW_PAGE                        *
            *             THIS ROUTINE ADDS ONE TO LINE_COUNT AND IF THE *
            *  MAXIMUM NUMBER OF LINE HAVE BEEN PRINTED ( MAXM = 60 )  *
            *  THEN PRINT A NEW PAGE AND RESET LINE_COUNT.             *
            ** * * * * * ************************************ * * * * * **
1810    MAY_NEW_PAGE:
            IF LINE_COUNT =) 50 THEN
               PAGE$ = SUM$(PAGE$,"1")
               PRINT #8, CHR$(12)
               PRINT #8, TAB(65);"PAGE ";PAGE$
               PRINT #8
               IF POSITION = 0 THEN
                  PRINT #8; TAB(6);"CODE";TAB(31);"DESCRIPTION";TAB(73);"VALUE"
                  PRINT #8; TAB(6);"----";TAB(31);"-----------";TAB(73);"-----"
               ELSE
                  PRINT #8; TAB(6);"ID";TAB(14);"CODE";TAB(36);"DESCRIPTION";TAB(73);"VALUE"
                  PRINT #8; TAB(6);"--";TAB(14);"----";TAB(36);"-----------";TAB(73);"-----"
               END IF
               LINE_COUNT = 4
            ELSE
               LINE_COUNT = LINE_COUNT + 1
            END IF
     PAGE_RETURN:
            RETURN
1900    REM ** * * * * * ************************************ * * * * * **
            *                          FIND_METHOD                       *
            ** * * * * * ************************************ * * * * * **
1910    FIND_METHOD:
            FOUND = 0
1920        !*****************************************************************
            !*     CHECK THE ELEMENT ALREADY IN THE ARRAY METHOD_VECTOR*
            !*   FOR THE NEW PARAMTER CODE "PARCODE".                      *
            !*****************************************************************
            FOR JF = 1 TO M_POS - 1
```

```
            IF PARCODE = METHOD_VECTOR(JF,2) THEN
               FOUND = 1
            END IF
         NEXT JF
      FIND_METHOD_RETURN:
         RETURN
2000  REM ** * * * * * ************************************ * * * * * **
         *                    REPORT_TYPE_1                            *
         *          REPORT THE DATA IN THE FOLLOWING FORMAT            *
         *               SPCODE   DESCRIPTION  VALUE                   *
         *                ----      -------     ---                    *
         *                ----      -------     ---                    *
         *                ----      -------     ---                    *
         *                MMM-      COUNTS                             *
         ** * * * * * ************************************ * * * * * **
2010   REPORT_TYPE_1:
         PRINT #8
         GOSUB MAY_NEW_PAGE
         WHILE SAMPLE = GET_SAM
2015        !************************************************
            !*    CHECK TO SEE IF PARCODE ALREADY STORED    *
            !************************************************
2020        GOSUB FIND_METHOD
            IF FOUND = 0 THEN
               METHOD_VECTOR(M_POS,2) = PARCODE
               M_POS = M_POS + 1
            END IF
            IF SEG$(PARCODE,1,2) <> "MM" OR SEG$(PARCODE,1,3) = "MMM" THEN
               METHOD_VECTOR(M_POS - 1,3) = PVAL
            ELSE
               GET #5, KEY #0 EQ SPCODE
               PRINT #8, TAB(5);SPCODE;TAB(19);SPEC2;TAB(73);PVAL
               GOSUB MAY_NEW_PAGE
            END IF
         !     METHOD_VECTOR(M_POS - 1,3) = "       "
2025        GET #1
         NEXT
2026     IF M_POS < 2 THEN
           GOTO REPORT_TYPE_1_RETURN
         END IF
         SEE_FLAG = 0
2028     REM ************************************************
            *  PRINT OUT THE METHOD CODES IN A N ORDER      *
            * 1 TIME PRINT OUT THE MM METHOD CODES          *
            * 2 TIME PRINT OUT THE M METHOD CODES           *
            * 3 TIME PRINT OUT THE MMM CODES                *
            ************************************************
2029     FOR RF = 1 TO 3
         FOR JR = 1 TO M_POS - 1
            MCODE$ = METHOD_VECTOR(JR,2)
2030        GET #6, KEY #0 EQ MCODE$
2040        SELECT RF
              CASE = 2
               IF SEG$(MCODE$,1,2) <> "MM" THEN
                  PRINT #8, TAB(5);BICODE;TAB(19);DESCR;TAB(73);METHOD_VECTOR(JR,3)
                  GOSUB MAY_NEW_PAGE
               END IF
              CASE = 1
               IF SEG$(MCODE$,1,2) = "MM" AND SEG$(MCODE$,1,3) <> "MMM"THEN
                  PRINT #8, TAB(5);BICODE;TAB(19);DESCR;TAB(73);METHOD_VECTOR(JR,3)
                  GOSUB MAY_NEW_PAGE
               END IF
              CASE = 3
```

```
            IF SEG$(MCODE$,1,3) = "MMM" THEN
              PRINT #8, TAB(5);BICODE;TAB(19);DESCR;TAB(73);METHOD_VECTOR(JR,3)
              GOSUB MAY_NEW_PAGE
            END IF
          END SELECT
        NEXT JR
        PRINT #8
        GOSUB MAY_NEW_PAGE
      NEXT RF
    REPORT_TYPE_1_RETURN:
      RETURN
2200  REM ** * * * * * * ************************************ * * * * * **
        *                      REPORT_TYPE_2                            *
        *           REPORT THE DATA IN THE FOLLOWING FORMAT             *
        *    ID     SPECIES CODE       DESCRIPTION          VALUE      *
        *    1         AF1010          FISH                            *
        *    1         M1020           SEX                   MALE      *
        *    1         M1120           AGE                    12       *
        *    -         ----            ---                   ----      *
        *             METHOD CODE      ---                   ----      *
        *    2         AF1010          FISH                            *
        *    2         M1020           SEX                   FEMALE    *
        *    -         -----           ---                   ------    *
        *    -         -----           ---                   ------    *
        *    1         AF1111          INSECTS                         *
        *    -         -----            -----                -------   *
        *      ETC.                                                     *
        ** * * * * * * ************************************ * * * * * **
2210  REPORT_TYPE_2:
      SORT_TABLE(0,5) = "^" \ SORT_TABLE(0,4) = "0" \ ROWW = 1
      M_POS = 1
      GOSUB SORT_RECORDS
      !*****************************************************************
      !*  GO THROUGH THE LINK LIST TABLE UNTIL "^" IS FOUND            *
      !*****************************************************************
      BEGIN = VAL%(SORT_TABLE(0,5))
      WHILE 1=1
2232    GET #5, KEY #0 EQ SORT_TABLE(BEGIN,1)
        IF SORT_TABLE(BEGIN,4) <> "0" THEN
          PRINT #8
          GOSUB MAY_NEW_PAGE
          PRINT #8, TAB(6);SORT_TABLE(BEGIN,4);TAB(13);SPCODES;TAB(24);SPEC2
          GOSUB MAY_NEW_PAGE
        END IF
2234    GET_IDNUM = SORT_TABLE(BEGIN,4)
        !*****************************************************************
        !*      PRINT OUT THE GROUPED SPECIES AND ID NUMBER              *
        !*   TOGATHER                                                    *
        !*****************************************************************
        WHILE  GET_IDNUM = SORT_TABLE(BEGIN,4)
2235      GET #6, KEY #0 EQ SORT_TABLE(BEGIN,2)
2237      PCODE$ = SORT_TABLE(BEGIN,2)
          !*****************************************************************
          !*    IF METHOD CODE = JUST AND M THEN STORED IT SO THAT         *
          !*  IT CAN BE PRINT OUT LATER                                    *
          !*****************************************************************
          IF VAL%(SORT_TABLE(BEGIN,4)) = 0 OR (SEG$(PCODE$,1,1) = "M" AND SEG$(PCODE$,1,2) <>
            METHOD_VECTOR(M_POS,1) = SORT_TABLE(BEGIN,4)
            METHOD_VECTOR(M_POS,2) = SORT_TABLE(BEGIN,2)
            METHOD_VECTOR(M_POS,3) = SORT_TABLE(BEGIN,3)
            M_POS = M_POS + 1
          ELSE
            PRINT #8, TAB(6);SORT_TABLE(BEGIN,4);TAB(13);BICODE;TAB(24);DESCR;TAB(73);SORT_T
```

```
                GOSUB MAY_NEW_PAGE
              END IF
              !*************************************************************
              !*      GET THE NEXT ROW IN THE LINK LIST ARRAY              *
              !*************************************************************
2240           BEGIN = VAL%(SORT_TABLE(BEGIN,5))
2250          NEXT
              GET_IDNUM = SORT_TABLE(BEGIN,4)
             !*************************************************************
             !*              PRINT OUT THE ONE M METHOD CODES             *
             !*************************************************************
              FOR TH = 1 TO M_POS
                 PCODE$ = METHOD_VECTOR(TH,2)
2255  IF VAL%(METHOD_VECTOR(TH,1)) <> 0 AND SEG$(PCODE$,1,1) = "M" AND SEG$(PCODE$,1,2) <> "MM" THEN
                 GET #6, KEY #0 EQ METHOD_VECTOR(TH,2)
                 PRINT #8, TAB(13);METHOD_VECTOR(TH,2);TAB(24);DESCR;TAB(73);METHOD_VECTOR(TH,3)
                 GOSUB MAY_NEW_PAGE
              END IF
2257         NEXT TH
            NEXT
            !*************************************************************
            !*      PRINT OUT THE REST OF THE METHOD CODES AT THE END OF *
            !*  THE SHEET                                                *
            !*************************************************************
2258         IF M_POS < 2 THEN
                GOTO REPORT_TYPE_2_RETURN
            END IF
            PRINT #8
            GOSUB MAY_NEW_PAGE
            FOR JR = 1 TO M_POS - 1
2260           GET #6, KEY #0 EQ METHOD_VECTOR(JR,2)
               IF VAL%(METHOD_VECTOR(JR,1)) = 0 THEN
                  PRINT #8, TAB(13);METHOD_VECTOR(JR,2);TAB(24);DESCR;TAB(73);METHOD_VECTOR(JR,3)
                  GOSUB MAY_NEW_PAGE
               END IF
2270        NEXT JR
2280     REPORT_TYPE_2_RETURN:
            RETURN
2300  REM ** * * * * * ********************************** * * * * * **
         *                        SORT_RECORDS                        *
         ** * * * * * ********************************** * * * * * **
2310  SORT_RECORDS:
2320    WHILE SAMPLE = GET_SAM
2370    NOT_FOUND = 1
        LAST = 0
2380    CONNTIN = VAL%(SORT_TABLE(0,5))
2440    WHILE NOT_FOUND = 1 AND SAMPLE = GET_SAM
2450      IF VAL%(SORT_TABLE(CONNTIN,4)) <> VAL%(IDNUM) THEN
             SORT_TABLE(ROWW,1) = SPCODE
             SORT_TABLE(ROWW,2) = PARCODE
             SORT_TABLE(ROWW,3) = PVAL
             SORT_TABLE(ROWW,4) = IDNUM
             SORT_TABLE(ROWW,5) = SORT_TABLE(LAST,5)
             SORT_TABLE(LAST,5) = STR$(ROWW)
             ROWW = ROWW + 1
             NOT_FOUND = 0
          ELSE
            !*****************************************************
            !*         GET THE NEXT ROW ELEMENTS                 *
            !*****************************************************
             LAST = CONNTIN
             CONNTIN = VAL%(SORT_TABLE(CONNTIN,5))
          END IF
```

```
2500    NEXT
2510    GET #1
        NEXT
2520 SORT_RECORDS_RETURN:
     RETURN
2999 REM ******************************************************
     ! *              EDIT SAMPLE DATA                        *
     ! *  THIS ROUTINE DISPLAYS A MENU WHICH INFORMS THE      *
     ! *  USER OF THE AVAILABLE OPTIONS .ONCE A CHOICE        *
     ! *  IS MADE A SUITABLE SCREEN WILL DISPLAY THE          *
     ! *  CURRENT RECORD VALUES .                             *
     ! ******************************************************
     !
     !
3000 EDIT_SAM:
3010  PRINT  CLRSCREEN
      PRINT  TAB(8);TITLE;LFEED
      PRINT  TAB(30);"SAMPLE EDITING MENU"
      PRINT  TAB(28);"----------------------";LFEED;LFEED
3020 PRINT  TAB(26);"1) EDIT SAMPLING INFORMATION"
      PRINT  TAB(26);"2) EDIT CODES OR VALUES"
      PRINT  TAB(26);"3) RETURN TO MAIN MENU"
      INPUT  "          CHOICE :";CHOICE
3030 IF CHOICE ( 1 OR CHOICE ) 3 THEN
      PRINT BELL
      GOTO 3020
3040 ON CHOICE GOSUB EDSAM_INFO,EDSAM_CODES,MAIN_MENU
      GOTO EDIT_SAM
      !          ************ END EDIT_SAM *************
      !
      ! ******************************
      ! * EDIT SAMPLING INFORMATION *
      ! ******************************
3100 EDSAM_INFO:
3110 L1="-------------------------------------------------------------------"
     L2="! SAMPLE # !   STATION #   ! SUBID !    SDATE    ! STIME ! TZONE !"
     L3="! SAMDES !"
     L4="         1) STATION #    2) SUBID     3) SDATE              4) STIME"
     L5="         5) TZONE        6) SAMDES    7) ANOTHER SAMPLE     8) EXIT"
3130 PRINT  CLRSCREEN;LFEED;LFEED
3140 INPUT "     ENTER SAMPLE # TO LOCATE";SAMIN
      IF LEN(STR$(SAMIN))(6 THEN
       PRINT BELL
       PRINT "     INVALID SAMPLE NUMBER"
       GOTO 3140
      ! ***************************************
      ! * GET SAMPLE AND DISPLAY INFORMATION *
      ! ***************************************
3150 RESET #1%
      GET #1%, KEY #0% EQ SAMIN
3160 PRINT  CLRSCREEN
      PRINT  TAB(8);TITLE;LFEED
3170 PRINT  TAB(9);L1
      PRINT  TAB(9);L2
      PRINT  TAB(9);L1
      PRINT  TAB(9);"! ";SAMPLE;TAB(19);" ! ";STATION;TAB(33);"  ! ";&
      SUBID;TAB(42);"  ! ";SDATE;TAB(57);" ! ";STIME;TAB(64);"  !  ";TZONE;&
      TAB(72);"  !"
      PRINT  TAB(9);L1
      PRINT  LFEED;LFEED
      PRINT  TAB(9);L1
      PRINT  TAB(9);L3;"  ";SAMDES
      PRINT  TAB(9);L1;LFEED
```

```
        PRINT  L4;LFEED
        PRINT  L5;LFEED;LFEED
3180 INPUT "     CHOICE :";CHOICE
        IF CHOICE<1 OR CHOICE>8 THEN
          PRINT BELL;"      =====>INVALID RESPONSE<====="
          SLEEP 3
          PRINT MOV+"17;30H";ERASE
          GOTO 3180
        END IF
        ON CHOICE GOTO 3200,3250,3300,3350,3400,3450,3130,3500
        !   ******************************************
        !   * DISPLAY PROMPTS FOR ENTERING NEW DATA *
        !   ******************************************
3200 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD STATION # IS ";STATION;LFEED
        INPUT "      ENTER THE NEW STATION # ";STATION1
        IF LEN(STATION1)<> 12 THEN PRINT BELL;"      INVALID STATION #"
          SLEEP 3
          GOTO 3200
        ELSE
          FIND #2%,KEY #0 EQ STATION1
          END IF
        STATION=STATION1
        UPDATE #1%
        GOTO 3160
3250 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD SUBID IS ";SUBID;LFEED
        INPUT "      ENTER THE NEW SUBID ";SUBID1
        IF LEN(SUBID1)<>4 THEN PRINT BELL
          PRINT "   ===> INCORRECT # OF CHARACTERS <===="
          SLEEP 3
          GOTO 3250
        END IF
        SUBID=SUBID1
        UPDATE #1%
        GOTO 3160
3300 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD SDATE IS ";SDATE;LFEED
        INPUT "      ENTER THE NEW SDATE '02-JUN-1985' ";SDATE1
        MFLAG=0
        SDAY=SEG$(SDATE1,1,2)
        SMON=SEG$(SDATE1,4,6)
        SYR=SEG$(SDATE1,8,11)
        IF VAL%(SDAY)<1 OR VAL%(SDAY)>31 THEN 3300 \ END IF
        FOR I=1 TO 12
          IF SMON=MONTH(I) THEN MFLAG=1 \ END IF
          IF MFLAG=1 THEN MFLAG=0
            GOTO 3310
          END IF
        NEXT I
        PRINT BELL;"     INVALID MONTH IN SAMPLING DATE"
        SLEEP 3
        GOTO 3300
3310 IF LEN(SYR)<>4 THEN PRINT BELL;"===> INVALID YEAR IN SAMPLING DATE <==="
          SLEEP 3
          GOTO 3300
        END IF
        SDATE=SDATE1
        UPDATE #1%
        GOTO 3160
3350 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD SAMPLING TIME IS ";STIME;LFEED
        INPUT "      ENTER THE NEW TIME ";STIME1
```

```
        IF LEN(STIME1)()4 THEN PRINT BELL;" ===) INCORRECT # OF CHARACTERS (==="
           SLEEP 3
           GOTO 3350
        END IF
        STIME=STIME1
        UPDATE #1%
        GOTO 3160
3400 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD TIME ZONE IS ";TZONE;LFEED
        INPUT "      ENTER THE NEW TIME ZONE ";TZONE1
        IF LEN(TZONE1)()3 THEN PRINT BELL;"==) INCORRECT LENGTH (=="
          SLEEP 3
          GOTO 3400
        END IF
        TZONE=TZONE1
        UPDATE #1%
        GOTO 3160
3450 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD SAMPLE DESCRIPTION IS ";SAMDES;LFEED
        PRINT "      ENTER THE NEW SAMPLE DESCRIPTION 'MAX 40 CHARS'"
        INPUT "     ";SAMDES1
        IF LEN(SAMDES1))40 THEN PRINT BELL;" ===) DESCRIPTION TOO LONG (==="
          SLEEP 3
          GOTO 3450
        END IF
        SAMDES=SAMDES1
        UPDATE #1%
        GOTO 3160
3500 CLOSE #2%
        RETURN
        !            ********** END EDSAM_INFO ***********
        !
        ! ************************************
        ! * EDIT CODES,ID NUMBER AND VALUES *
        ! ************************************
4000 EDSAM_CODES:
4010 L1="-------------------------------------------------"
     L2="|  SPCODE  |  IDNUM  |   PARCODE   |   VALUE   |"
     L4="          1) SPCODE      2) IDNUM      3) PARCODE      4) VALUE"
     L5="          5) ANOTHER SAMPLE                  6) NEXT MATCH "
     L6="          7) SELECT ANOTHER CODE    8) DELETE RECORD    9) EXIT"
4020 PRINT CLRSCREEN;LFEED;LFEED
        CHFLAG = 0
        INPUT "     ENTER SAMPLE # TO LOCATE ";SAMIN
        IF LEN(STR$(SAMIN))()6 THEN
          PRINT BELL;"       INVALID SAMPLE NUMBER"
          GOTO 4020
4030 RESET #1%
        GET #1%,KEY #0 EQ SAMIN
        RESET #1%
        PRINT LFEED
4035 PRINT CLRSCREEN;LFEED
        PRINT  "     EDITING SAMPLE NUMBER - ";SAMPLE
        PRINT  LFEED;LFEED
        PRINT   "     KEY ON --  1) SPECIES CODE"
        PRINT  "                2) PARAMETER CODE"
        INPUT  "                3) ID NUMBER ";RES
        PRINT  CLRSCREEN
        RES1=RES
        IF RES="1" THEN
          INPUT"     ENTER 6 CHARACTER SPECIES CODE ";INCODE
          GET #1,KEY #0 EQ SAMIN
          WHILE SAMPLE=SAMIN
```

```
          IF SPCODE=INCODE THEN 4040 \ END IF
          GET #1
         NEXT
         PRINT "    SPECIES CODE ";INCODE;" NOT ON SAMPLE ";SAMIN
         SLEEP 3
         GOTO 4020
       ELSE
       IF RES="2" THEN
         INPUT"     ENTER 6 CHARACTER PARAMETER CODE ";INCODE
         GET #1,KEY #0 EQ SAMIN
         WHILE SAMPLE = SAMIN
          IF PARCODE=INCODE THEN 4040
          END IF
         GET #1
         NEXT
         PRINT "    PARAMETER CODE ";INCODE;" NOT FOUND IN SAMPLE ";SAMIN
         SLEEP 3
         GOTO 4020
       END IF
       END IF
4036 IF RES="3" THEN
       INPUT "      ENTER ID NUMBER TO LOCATE ";IDNUM1
       GET #1,KEY #0 EQ SAMIN
       WHILE SAMPLE = SAMIN
         IF IDNUM=IDNUM1 THEN 4040
         END IF
         GET #1
       NEXT
       PRINT "      ID NUMBER ";IDNUM1;" NOT FOUND IN SAMPLE ";SAMIN
       SLEEP 3
       GOTO 4020
     END IF
     !  ****************************************
     !  * DISPLAY SAMPLE RECORD WITH CHOICES *
     !  ****************************************
4040   PRINT CLRSCREEN
       PRINT  TAB(8);TITLE;LFEED
4050 PRINT  TAB(16);L1
       PRINT   TAB(16);L2
       PRINT  TAB(16);L1
       PRINT  TAB(16);"|  ";SPCODE;TAB(25);"  |  ";IDNUM;TAB(35);"  |   ";&
       PARCODE;TAB(47);"    |  ";PVAL;TAB(60);"  |"
       PRINT  TAB(16);L1
       PRINT  LFEED;LFEED
       PRINT  L4;LFEED
       PRINT  L5;LFEED
       PRINT  L6;LFEED
4060 INPUT "      CHOICE: ";CHOICE
       IF CHOICE(1 OR CHOICE )9 THEN
         PRINT BELL;BELL;"====) INVALID RESPONSE (===="
         SLEEP 3
         GOTO 4040
       END IF
       IF CHOICE =) 5  AND CHFLAG = 1 THEN
          UPDATE #1%
       END IF
       CHFLAG=0
       ON CHOICE GOTO 4100,4150,4200,4250,4020,4300,4035,4500,4400
       ! ******************************************
       ! * DISPLAY PROMPTS FOR ENTERING NEW DATA *
       ! ******************************************
       !
       ! **************************
```

```
        ! * CHANGING SPECIES CODE *
        ! ***************************
4100 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD SPECIES CODE IS ";SPCODE;LFEED
        INPUT "     ENTER THE SPECIES CODE ";SPCODE1
        IF LEN(SPCODE1)()6 THEN
          PRINT BELL;" ===) INCORRECT # OF CHARACTERS (==="
          SLEEP 3
          GOTO 4100
        END IF
        SPCODE=SPCODE1
        CHFLAG=1
        GOTO 4040
        ! *************************
        ! * CHANGING  ID NUMBER *
        ! *************************
4150 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD ID NUMBER IS ";IDNUM;LFEED
        INPUT "     ENTER THE NEW ID NUMBER ";IDNUM1
        IF LEN(IDNUM1))4 THEN PRINT BELL;"====) INCORRECT # OF CHARACTERS (===="
          SLEEP 3
          GOTO 4150
        END IF
        IDNUM=IDNUM1
        CHFLAG=1
        GOTO 4040
        ! ******************************
        ! * CHANGING PARAMETER CODE *
        ! ******************************
4200 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD CODE IS ";PARCODE;LFEED
        INPUT "     ENTER THE CODE ";PARCODE1
        IF LEN(PARCODE1)()6 THEN PRINT BELL;" ===) INVALID CODE (==="
          SLEEP 3
          GOTO 4200
        END IF
        PARCODE=PARCODE1
        CHFLAG=1
        GOTO 4040
        ! ********************
        ! * CHANGING VALUE *
        ! ********************
4250 PRINT MOV+"17;30H";ERASE
        PRINT TAB(5);"THE OLD VALUE IS ";PVAL;LFEED
        INPUT "     ENTER THE NEW VALUE ";PVAL1
        IF LEN(PVAL1))7 THEN PRINT BELL;"===) TOO MANY CHARACTERS (==="
          SLEEP 3
          GOTO 4250
        END IF
        PVAL=PVAL1
        CHFLAG=1
        GOTO 4040
        ! ****************************
        ! * LOCATING THE NEXT MATCH *
        ! ****************************
4300 WHILE SAMPLE = SAMIN
        IF RES1="1" THEN
        GET #1
          IF SPCODE=INCODE THEN 4040
          END IF
        END IF
        IF RES1="2" THEN
          GET #1
```

```
          IF PARCODE=INCODE THEN 4040
          END IF
        END IF
        IF RES1="3" THEN
          GET #1
          IF IDNUM=IDNUM1 THEN 4040
          END IF
        END IF
      NEXT
4400 RETURN
4500    ! ********************************
        ! *       DELETE A RECORD        *
        ! ********************************
4510  DELETE #1
      GOTO 4035
        !       ************ END  EDSAM_CODES **************
        !
        ! ****************************
        ! *  EDIT DESCRIPTION FILES  *
        ! ****************************
5000 EDIT_DES:
5010 L1="-------------------------------------------------------------------------------"
5015 L2="!  SPECIES CODE    !                  DESCRIPTION                 !"
5020 L3="!  REFERENCE  !  "
5030 L4="!  PARAMETER CODE  !                  DESCRIPTION                 !"
5040 L5="     1) DESCRIPTION          2) REFERENCE         3) EXIT"
5050 L6="            1) DESCRIPTION          2) EXIT"
5100 OPEN"SPECIES.NEW" AS FILE #3,&
          ORGANIZATION INDEXED FIXED,&
          PRIMARY KEY SCODE,&
          ALTERNATE KEY DES DUPLICATES,&
          ACCESS MODIFY,MAP RECTHR,ALLOW MODIFY
     OPEN"BIOCODE.NEW" AS FILE #4,&
          ORGANIZATION INDEXED FIXED,&
          PRIMARY KEY BICODE,&
          ACCESS MODIFY,MAP RECFOR,ALLOW MODIFY
       ! ************************************************
       ! *   CREATE SCREEN FOR DESCRIPTION FILE EDITING  *
       ! ************************************************
5110 PRINT CLRSCREEN;LFEED
       PRINT  TAB(8);TITLE
       PRINT  LFEED;LFEED
       PRINT  TAB(26);"CODE DESCRIPTION FILE EDITOR"
       PRINT  LFEED
       PRINT  TAB(30);"1) EDIT SPECIES.NEW"
       PRINT  TAB(30);"2) EDIT BIOCODE.NEW"
       PRINT  TAB(30);"3) ADD A RECORD TO SPECIES.NEW"
       PRINT  TAB(30);"4) ADD A RECORD TO BIOCODE.NEW"
       PRINT  TAB(30);"5) DELETE A RECORD FROM SPECIES.NEW"
       PRINT  TAB(30);"6) DELETE A RECORD FROM BIOCODE.NEW"
       PRINT  TAB(30);"7) RETURN TO MAIN MENU"
       PRINT  LFEED;LFEED
       INPUT   "          CHOICE: ";CHOICE
          IF CHOICE=7 THEN
            CLOSE #3
            CLOSE #4
5140 ON CHOICE GOTO EDIT_SPECIES,EDIT_BIOCODE,ADD_SPECIES,ADD_BIOCODE,&
       DEL_SPECIES,DEL_BIOCODE,MAIN_MENU
5200 RETURN
       !          ********** END EDIT_DES **********
       !
5500 EDIT_SPECIES:
       PRINT CLRSCREEN
```

```
        PRINT  CLRSCREEN
        PRINT  TAB(8);TITLE;LFEED;LFEED
5510 INPUT   "              ENTER 6 CHARACTER SPECIES CODE ";INCODE
5520 GET #3,KEY #0 EQ INCODE
5530 PRINT CLRSCREEN;LFEED
        PRINT  TAB(8);TITLE;LFEED
        PRINT  TAB(10);L1
        PRINT  TAB(10);L2
        PRINT  TAB(10);L1
        PRINT  TAB(10);"¦";TAB(17);SCODE;"       ¦ ";DESCRIP;TAB(73);"¦"
        PRINT  TAB(10);L1
        PRINT  LFEED
        PRINT  TAB(10);L1
        PRINT  TAB(10);L3;REFER;TAB(73);"¦"
        PRINT  TAB(10);L1
        PRINT  LFEED
        PRINT  TAB(10);L5
        PRINT  LFEED
5540 INPUT   "              CHOICE:";CHOICE
        IF CHOICE(1 OR CHOICE ) 3 THEN
           PRINT BELL;"             INVALID RESPONSE"
           GOTO 5540
        END IF
        IF CHOICE=3 THEN GOTO 5620 \ END IF
     !*******************************
     !* EDIT SPECIES  DESCRIPTION *
     !*******************************
5550 IF CHOICE=1 THEN
        PRINT "          TYPE NEW DESCRIPTION (MAX 40 CHARACTERS)"
        INPUT "   NEW DESCRIPTION - ";DESC1
        IF LEN(DESC1) ) 40 THEN
           PRINT BELL;"INCORRECT NUMBER OF CHARACTERS"
           PRINT MOV + "9;14H";ERASE
           GOTO 5550
        END IF
        DELETE #3%
        DES = SEG$(DESC1,1,4)
        DES1 = SEG$(DESC1,5,40)
        PUT #3%
        PRINT "      NEW DESCRIPTION HAS BEEN ENTERED ON SPECIES.NEW"
        SLEEP 3
     ELSE
        IF CHOICE=2 THEN 5600
     END IF
5570 INPUT "        DO YOU WISH TO CHANGE MORE SPECIES DESCRIPTIONS (Y/N)";RES
        IF RES="Y" THEN 5500 \ END IF
        IF RES="N" THEN 5110 \ END IF
        PRINT BELL;"      INCORRECT RESPONSE"
           GOTO 5570
     !*******************************
     !*      EDIT REFERENCE       *
     !*******************************
5600 PRINT "          TYPE NEW REFERENCE (MAX 40 CHARACTERS)"
        INPUT "   NEW REFERENCE - ";REFER
          IF LEN(REFER))40 THEN
             PRINT BELL;"    TOO MANY CHARACTERS      "
             GOTO 5600
          END IF
5610 UPDATE #3%
        PRINT "         SPECIES.NEW HAS BEEN UPDATED "
        SLEEP 3
5620 INPUT"     DO YOU WISH TO EDIT MORE REFERENCES ? (Y/N) ";RES
        IF RES="Y" THEN 5500 \ END IF
```

```
        IF RES="N" THEN 5110 \ END IF
        PRINT BELL;"       INCORRECT RESPONSE "
        GOTO 5620
5650            !*********    END OF EDIT_SPECIES    ***********
   !
        ! *********************
        ! * EDIT BIOCODE.NEW *
        ! *********************
5700 EDIT_BIOCODE:
        PRINT CLRSCREEN
        PRINT TAB(8);TITLE;LFEED;LFEED
        INPUT "              ENTER 6 CHARACTER CODE ";INCODE
5710 GET #4,KEY #0 EQ INCODE
        PRINT CLRSCREEN;LFEED
        PRINT TAB(8);TITLE;LFEED
        PRINT TAB(10);L1
        PRINT TAB(10);L4
        PRINT TAB(10);L1
        PRINT TAB(10);"!     ";BICODE;"     !  ";DESCR;TAB(60);"!"
        PRINT TAB(10);L1
        PRINT LFEED;LFEED
        PRINT TAB(10);L6
        PRINT LFEED;LFEED
5720 INPUT "               CHOICE: ";CHOICE
        IF CHOICE(1 OR CHOICE)2 THEN
           PRINT BELL;"  INVALID RESPONSE "
           GOTO 5720
        END IF
        IF CHOICE=2 THEN 5110 \ END IF
5730 IF CHOICE=1 THEN
        PRINT "    TYPE NEW DESCRIPTION (MAX 50 CHARACTERS)"
        INPUT "         NEW DESCRIPTION ";DESCR
5740 UPDATE #4%
        PRINT "      BIOCODE.NEW HAS NOW BEEN UPDATED "
        SLEEP 3
5750 INPUT "     DO YOU WISH TO EDIT MORE PARAMETER CODES ? (Y/N)";RES
        IF RES="Y" THEN 5700 \ END IF
        IF RES="N" THEN 5110
        ELSE
         PRINT BELL;"    INVALID RESPONSE"
         GOTO 5750
    !          ********** END EDIT_BIOCODE ************
    !
        ! *******************************
        ! * ADD RECORD TO SPECIES.NEW *
        ! *******************************
5800 ADD_SPECIES:
5810 PRINT CLRSCREEN;LFEED
        PRINT TAB(8);TITLE;LFEED
        PRINT TAB(28);"ADD RECORD TO SPECIES.NEW";LFEED;LFEED
5820 INPUT "   TYPE 6 CHARACTER SPECIES CODE - ";INCODE
        IF LEN(INCODE)() 6 THEN
          PRINT BELL;"    INCORRECT NUMBER OF CHARACTERS "
          PRINT MOV+"7;5H";ERASE
          GOTO 5820
        END IF
        SCODE=INCODE
        PRINT LFEED
5830 INPUT "   TYPE DESCRIPTION (MAX 40 CHARACTERS) - ";DESC1
        IF LEN(DESC1))40 THEN
          PRINT BELL;"    INCORRECT NUMBER OF CHARACTERS"
          PRINT MOV+"9;5H";ERASE
          GOTO 5830
```

```
        END IF
        DESCRIP=DESC1
        PRINT LFEED
5840 INPUT "   TYPE REFERENCE (MAX 40 CHARACTERS) - ";REFER2
        IF REFER2="" THEN REFER=" NO REFERENCE " \ END IF
        IF LEN(REFER2))40 THEN
          PRINT BELL;"    INCORRECT NUMBER OF CHARACTERS"
          PRINT MOV+"11;5H";ERASE
          GOTO 5840
        END IF
        REFER=REFER2
5850 PUT #3
        PRINT LFEED;LFEED
        PRINT "       THIS RECORD HAS BEEN ADDED TO SPECIES.NEW"
        SLEEP 3
        PRINT CLRSCREEN;LFEED
5860 PRINT LFEED
        INPUT"       DO YOU WISH TO ADD MORE SPECIES CODES ? (Y/N)";RES
        IF RES="Y" THEN 5810
        ELSE
         GOTO 5110
        END IF
     !       ********** END OF ADD_SPECIES **********
     !
        ! ******************************
        ! * ADD RECORD TO BIOCODE.NEW *
        ! ******************************
5900 ADD_BIOCODE:
        PRINT CLRSCREEN;LFEED
        PRINT TAB(8);TITLE;LFEED
        PRINT TAB(28);"ADD RECORD TO BIOCODE.NEW";LFEED;LFEED
5910 INPUT"          TYPE 6 CHARACTER CODE - ";INCODE
        IF LEN(INCODE)()6 THEN
          PRINT BELL;"    INCORRECT NUMBER OF CHARACTERS"
          PRINT MOV+"7;5H";ERASE
          GOTO 5910
        END IF
        BICODE=INCODE
        PRINT LFEED
5920 PRINT"          TYPE CODE DESCRIPTION (MAX 50 CHARACTERS) - "
        INPUT DESC1
        IF LEN(DESC1))50 THEN
          PRINT BELL;"     INCORRECT NUMBER OF CHARACTERS"
          PRINT MOV+"9;5H";ERASE
          GOTO 5920
        END IF
        DESCR=DESC1
        PRINT LFEED
5930 PUT #4
        PRINT LFEED;LFEED
        PRINT "      THIS RECORD HAS BEEN ADDED TO BIOCODE.NEW"
        SLEEP 3
        PRINT CLRSCREEN;LFEED;LFEED
5940 INPUT " DO YOU WISH TO ADD MORE CODES TO BIOCODE.NEW ? (Y/N)";RES
        IF RES="Y" THEN 5900
        ELSE
        GOTO 5110
        END IF
     !        ********** END ADD_BIOCODE **********
     !
        ! ************************************
        ! * DELETE RECORD FROM SPECIES.NEW *
        ! ************************************
```

```
6000 DEL_SPECIES:
       PRINT CLRSCREEN;LFEED;LFEED
       PRINT TAB(25);"DELETE RECORD FROM SPECIES.NEW"
       PRINT LFEED;LFEED
6010 INPUT "     TYPE SPECIES CODE TO BE DELETED ";INCODE
       GET #3%,KEY #0 EQ INCODE
       DELETE #3%
6020 PRINT LFEED;LFEED
       PRINT "     SPECIES CODE - ";INCODE;" - HAS BEEN DELETED"
       PRINT LFEED;LFEED;LFEED
       INPUT "     DO YOU WISH TO DELETE MORE SPECIES CODES ? (Y/N) ";RES1
6030 IF RES1="Y" THEN 6000
6040 GOTO 5110
     !       ********** END DEL_SPECIES **********
     !
       ! ***********************************
       ! * DELETE RECORD FROM BIOCODE.NEW *
       ! ***********************************
6500 DEL_BIOCODE:
       PRINT CLRSCREEN;LFEED;LFEED
       PRINT TAB(25);"DELETE RECORD FROM BIOCODE.NEW"
       PRINT LFEED;LFEED
6510 INPUT"     TYPE CODE TO BE DELETED ";INCODE
       GET #4%,KEY #0 EQ INCODE
       DELETE #4%
6520 PRINT "    CODE - ";INCODE;" - HAS BEEN DELETED FROM BIOCODE.NEW"
       PRINT LFEED;LFEED;LFEED
       INPUT "     DO YOU WISH TO DELETE MORE CODES ? (Y/N) ";RES1
6530 IF RES1="Y" THEN 6500
       ELSE
       GOTO 5110
     END IF
     !       ********* END DEL_BIOCODE *********
     !
7000 ! ******************************************************
     ! *               DELETE A SAMPLE NUMBER               *
     ! *    DELETES A SAMPLE FROM THE FILE BIODAT.NEW       *
     ! ******************************************************
7010 DEL_SAM:
     I = 1
     PRINT CLRSCREEN;LFEED;LFEED
     PRINT  TAB(23);"B I O L O G I C A L   S Y S T E M "
     PRINT
     PRINT  MOV + "6H"
     PRINT  TAB(27);"D E L E T E   S A M P L E S"
     PRINT
     PRINT  TAB(27);" TO EXIT ROUTINE ENTER $ "
7020 PRINT  MOV + "12H"
     INPUT " ENTER SAMPLE NUMBER TO DELETE FROM BIODAT.DAT ";SAM$
     IF SAM$ = "$" THEN
        GOTO 7040
     END IF
7030 GET #1, KEY #0 EQ VAL%(SAM$)
     SAM_ARRAY(I) = SAM$
     I = I + 1
     GOTO 7020
7040 RESET #1
     FOR J = 1 TO I - 1
       PRINT MOV + "17H"
       PRINT "     DELETING THE SAMPLE ";SAM_ARRAY(J)
       GET #1, KEY #0 EQ VAL%(SAM_ARRAY(J))
       WHILE SAMPLE = VAL%(SAM_ARRAY(J))
         DELETE #1
```

```
        GET #1
      NEXT
    NEXT J
    RETURN
      ! **********************************
      ! * CLOSE FILES AND EXIT PROGRAM *
      ! **********************************
29000 LEAVE:
29010 CLOSE #1%
      CLOSE #2%
      CLOSE #3%
      CLOSE #4%
      CLOSE #5 \ CLOSE #6 \ CLOSE #7 \ CLOSE #8
      GOTO 32000
29900 REM ************************************
      ! *          ERROR ROUTINES          *
      ! ************************************
30000 IF ERR=155 AND ERL=525 THEN
      PRINT BELL;"SAMPLE # ";SAMIN;" NOT ON FILE"
      SLEEP 3
      RESUME 510
30010 IF ERR=155 AND ERL=140 THEN
      PRINT MOV+"17;30H"
      PRINT BELL;"STATION # NOT ON FILE "
      SLEEP 3
      PRINT MOV+"17;30H";ERASE
      RESUME 140
30015 IF ERR=155 AND ERL=3200 THEN
      PRINT BELL;"STATION NOT ON FILE"
      SLEEP 3
      RESUME 3200
30020 IF ERR=155 AND ERL=3150 THEN
      PRINT BELL;"SAMPLE NOT ON FILE "
      SLEEP 3
      RESUME 3130
30025 IF ERR=155 AND ERL=4030 THEN
      PRINT BELL;"  SAMPLE NOT ON FILE "
      SLEEP 3
      RESUME 4020
30030 IF ERR=11 AND ERL=4300 THEN
      PRINT MOV+"17;30H";ERASE
      PRINT BELL;" NO MATCH FOUND "
      SLEEP 3
      RESUME 3010
30040 IF ERR=155 AND ERL=5520 THEN
      PRINT BELL;MESS1
      SLEEP 3
      RESUME 5500
30050 IF ERR=155 AND ERL=5710 THEN
      PRINT BELL;MESS1
      SLEEP 3
      RESUME 5700
30060 IF ERR=155 AND ERL=6010 THEN
      PRINT BELL;MESS1
      SLEEP 3
      RESUME 6000
30070 IF ERR=155 AND ERL=6510 THEN
      PRINT BELL;MESS1
      SLEEP 3
      RESUME 6500
30075 IF ERR=134 AND ERL=5850 THEN
      PRINT BELL;MESS2
      SLEEP 3
```

```
        RESUME 5810
30080 IF ERR=134 AND ERL=5930 THEN
        PRINT BELL;MESS2
        SLEEP 3
        RESUME 5900
30085 IF ERL = 1233 THEN
        OPEN "PRTFILE.DAT" AS FILE #8, &
          ORGANIZATION SEQUENTIAL VARIABLE, &
          RECORDSIZE 132%, MAP PREC, ACCESS MODIFY
        RESUME 1240
30088 IF ERL = 2256 THEN
          RESUME 2258
      END IF
30090 IF ERR=145 THEN
        PRINT BELL;"   FIELD SIZE TOO LARGE ";ERL;" **";METHOD_VECTOR(JR,2);"**"
        SLEEP 3
        RESUME 100
      END IF
      IF ERR=11 AND ERL=4035 THEN
        PRINT BELL
        SLEEP 3
        RESUME 4040
30093 IF ERL = 2450 OR ERL = 2380 THEN
         SORT_TABLE(ROWW,1) = SPCODE
         SORT_TABLE(ROWW,2) = PARCODE
         SORT_TABLE(ROWW,3) = PVAL
         SORT_TABLE(ROWW,4) = IDNUM
         SORT_TABLE(ROWW,5) = SORT_TABLE(LAST,5)
         SORT_TABLE(LAST,5) = STR$(ROWW)
         ROWW = ROWW + 1
         NOT_FOUND = 0
         RESUME 2510
      END IF
30097 IF ERL = 7030 THEN
        PRINT MOV + "20H";"  ERROR IN SAMPLE NUMBER "
        SLEEP 2%
        RESUME 7020
      END IF
30100 SELECT ERL
        CASE = 130
          IF ERR = 155 THEN
             RESUME 135
          END IF
        CASE 1190,1186
          PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
          PRINT " ILLEGAL NUMBER ---- LAST NUMBER ENTERED CORRECTLY WAS ";SAM_ARRAY(I-1)
          SLEEP 1%
          RESUME 1120
        CASE = 1260
          PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
          PRINT " ILLEGAL NUMBER FOR PROVIDED "
          SLEEP 1%
          RESUME 1240
        CASE = 1360
          PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
          PRINT " HIT THE END OF FILE MARKER WHEN CHECKING THE IDS "
          SLEEP 1%
          RESUME 1375
        CASE = 2025
          PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
          PRINT " HIT THE END OF FILE MARKER AS PRINTING REPORT TYPE 1 "
          SLEEP 1%
          RESUME 2026
```

```
CASE = 2510
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT " HIT THE END OF FILE MARKER AS PRINTING REPORT TYPE 2 "
  SLEEP 3%
  RESUME 2520
CASE = 1350
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT " INVALID ID NUMBER FOUND ";IDNUM
  POSITION = 1
  SLEEP 2%
  RESUME 1360
CASE = 2237
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT " INVALID ID NUMBER FOUND ";IDNUM
  DESCR = "  ***      INVALID ID NUMBER ***"
  PRINT #8, TAB(6);IDNUM;TAB(24);DESCR
  GOSUB MAY_NEW_PAGE
  SLEEP 2%
  RESUME 2240
CASE = 1515
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT "    STATION NOT FOUND ON FILE ";STATION,ERR
  PRINT #8, "*******--)     STATION NOT FOUND ON FILE   (--*****";STATION
  PRINT #8, CHR$(12)
  SLEEP 1%
  RESUME 1400
CASE = 2210
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT "     INVALID NUMBER FOUND ---- NO DATA FOUND "
  SLEEP 2%
  RESUME 2280
CASE = 2232
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT "    SPECIES CODE NOT FOUND ON FILE ";SPCODE
  SPEC2 = "***   SPECIES CODE NOT FOUND ON FILE   ***"
  IDNUM = SORT_TABLE(BEGIN,4)
  SPCODE = SORT_TABLE(BEGIN,1)
  PRINT #8, TAB(6);IDNUM;TAB(13);SPCODE;TAB(24);SPEC2
  GOSUB MAY_NEW_PAGE
  SLEEP 1%
  RESUME 2234
CASE = 2235
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT "    PARAMETER CODE NOT FOUND ON FILE ";PARCODE
  DESCR = "***   PARAMETER CODE NOT FOUND ON FILE   ***"
  BICODE = SORT_TABLE(BEGIN,2)
  SLEEP 1%
  RESUME 2237
CASE = 2240
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT "     END OF THE LIST "
  SLEEP 1%
  RESUME 2258
CASE  2255
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
  PRINT "     CODE NOT FOUND ON FILE ";METHOD_VECTOR(JR,2)
  DESCR = "***   CODE NOT FOUND ON FILE   ***"
  PRINT #8, TAB(13);METHOD_VECTOR(JR,2);TAB(24);DESCR;TAB(73);METHOD_VECTOR(JR,3)
  GOSUB MAY_NEW_PAGE
  SLEEP 1%
  RESUME 2257
CASE  2260
  PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
```

```
            PRINT "     CODE NOT FOUND ON FILE ";METHOD_VECTOR(JR,2)
            DESCR = "***   CODE NOT FOUND ON FILE   ***"
            PRINT #8, TAB(13);METHOD_VECTOR(JR,2);TAB(24);DESCR;TAB(73);METHOD_VECTOR(JR,3)
            GOSUB MAY_NEW_PAGE
            SLEEP 1%
            RESUME 2270
          CASE = 2020
            PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
            PRINT "    SPECIES CODE NOT FOUND ON FILE ";SPCODE
            SPEC2 = "***   SPECIES CODE NOT FOUND ON FILE   ***"
            PRINT #8, TAB(5);SPCODE;TAB(19);SPEC2;TAB(73);PVAL
            GOSUB MAY_NEW_PAGE
            SLEEP 1%
            RESUME 2025
          CASE = 2030
            PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
            PRINT "     CODE NOT FOUND ON FILE ";METHOD_VECTOR(JR,2)
            DESCR = "***   CODE NOT FOUND ON FILE   ***"
            PVAL = "       "
            SLEEP 1%
            RESUME 2040
          CASE = 1430
            CLOSE #8
            KILL "PRTFILE.DAT"
            RESUME 100
          CASE ELSE
            PRINT CHR$(27) + "[22H",,,,,,,,ERASE1,CHR$(27) + "[21H"
            PRINT " ERROR NOT ACCOUNTED FOR ";ERR;" AT LINE ";ERL
            RESUME 32000
        END SELECT
32000 END
```

```
5 ON ERROR GOTO 30000
10      REM * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            *                                                       *
            *                    N A B I O D A T                    *
            *                                                       *
            *          BIOLOGICAL DATA MANAGEMENT SYSTEM            *
            *                                                       *
            *              B I O S H E E T . B A S                  *
            *                                                       *
            *     THIS PROGRAM IS DESIGNED TO PRODUCE TWO           *
            * DIFFERENT DATA SHEETS:                                *
            *                      1) FIELD SHEET                   *
            *                      2) CONTRACTOR'S SHEET            *
            *     THE PROGRAM OPERATES ON A MENU FOR THE            *
            * CHOICE OF SHEETS WANTED                               *
            *     THIS PROGRAM HAS THE FOLLOWING QUESTION           *
            * WHERE THE LIST ORDER IS THE ORDER THAT THEY           *
            * FOLLOW IN THE PROGRAM                                 *
            *                1) WHICH TYPE OF SHEET WOULD YOU       *
            *                     LIKE ?                            *
            *                2) HOW MANY DIFFERENT SHEETS WOULD     *
            *                     YOU LIKE ?                        *
            *                3) WHAT IS THE STATION NUMBER          *
            *                4) HOW MANY SHEETS WITH THE SAME       *
            *                     STATION NUMBER ON IT              *
            * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
60      MAP (RECFOUR) STRING WQSTN=12%,WRSTN=7%,STNDES=64%
70          OPEN "[LABS11]STNFIL.NEW" AS FILE #4%, &
                ORGANIZATION INDEXED FIXED, &
                PRIMARY KEY WQSTN, &
                ACCESS MODIFY, MAP RECFOUR, &
                ALLOW MODIFY
73      MAP (BIOREC) INTEGER SAMPLE, STRING STATION=12%,SUBID=4%,SDATE=11%, &
            STIME=4%,TZONE=3%,SAMDES=40%,SPCODE=6%,IDNUM=4%,PARCODE=6%,PVAL=7%
77      OPEN "[LABS11.BIO]BIODAT.NEW" FOR INPUT AS FILE #5, &
            ORGANIZATION INDEXED FIXED, &
            PRIMARY KEY SAMPLE DUPLICATES, &
            ALTERNATE KEY SPCODE DUPLICATES CHANGES, &
            ALTERNATE KEY IDNUM DUPLICATES CHANGES, &
            ALTERNATE KEY PARCODE DUPLICATES CHANGES, &
            ACCESS MODIFY, MAP BIOREC, ALLOW MODIFY
80      DECLARE STRING CLRSCREEN,UNDER_LINE,BLANK_LINE, VALUE_LINE, &
                    ERASE_LINE,WORD_LINE, INPUT_TABLE(50,2)
95      DECLARE INTEGER ANS
100     CLRSCREEN = CHR$(27) + "[5H" + CHR$(27) +"[2J"
101     MOV$ = CHR$(27) + "["
102     UNDER_LINE = "------------------------------------------------------------"
104     WORD_LINE = "__________________________________"
105     ERASE_LINE = CHR$(27) + "K"
106     VALUE_LINE = " __________ "
110     OPEN "TXCO:" FOR OUTPUT AS FILE #2%
115     PRINT #2,CHR$(27) + '?"#'
120     REM ** * * * * * ********************** * * * * * **
            *                  CREATE THE MENU                  *
            ** * * * * * ********************** * * * * * **
125     ANS$ = ""
130     PRINT CLRSCREEN;MOV$ + "5H"
140     PRINT "                        BIOLOGICAL FIELD SHEET MENU"
150     PRINT "                        ---------- ----- ----- ----"
160     PRINT "                      1 )  STRAIGHT FIELD SHEET"
170     PRINT "                      2 )  CONTRACTOR'S DATA SHEET"
175     PRINT MOV$ + "19H";"------------------------------ERROR MESSAGE AREA ----------------------
180     PRINT MOV$ + "12H",,,,,ERASE_LINE
```

```
185     PRINT MOV$ + "11H"
190     INPUT "            WHICH TYPE OF SHEET DO YOU WANT (1 OR 2) ";ANS$
200     ANS = VAL%(ANS$)
210     IF ANS ( 1 OR ANS ) 2 THEN
          GOTO 180
        END IF
        RESP = ANS
215     REM ** * * * * * ********************************** * * * * * **
            *    ASSIGN CERTAIN VALUES FOR THE DIFFERENT TYPES OF    *
            *  SHEETS                                                *
            ** * * * * * ********************************** * * * * * **
220     IF ANS = 1 THEN
          STRG_LEN = 12
          MESS1$ = "STATION "
          MESS$ = "F I E L D    S H E E T"
        ELSE
           STRG_LEN = 6
           MESS1$ = "SAMPLE "
           MESS$ = "C O N T R A C T O R ' S   D A T A   S H E E T"
        END IF
300     GOSUB INPUT_ROUTINE
305     REM ** * * * * * ********************************** * * * * * **
            *      ASK FOR THE NUMBER OF COPIES OF SHEETS WANTED     *
            ** * * * * * ********************************** * * * * * **
310     PRINT MOV$ + "16H",,,,,ERASE_LINE;MOV$ + "15H"
320     INPUT "                    HOW MANY COPIES DO YOU WANT ";ANS$
330     ANS = VAL%(ANS$)
        IF ANS ( 0 THEN
          GOTO 310
        END IF
340     FOR K = 1 TO VAL%(ANS$)
350     FOR J = 1 TO I
           IF RESP = 1 THEN
             GOSUB ADD_IN_FIELD_VALUES
           ELSE
             RESET #4
             RESET #5
             GOSUB ADD_IN_DATA_VALUES
           END IF
360     REM ** * * * * * ********************************** * * * * * **
            *                PRINT OUT THE BUILD SHEET               *
            ** * * * * * ********************************** * * * * * **
365       GOSUB PRINT_SHEETS
        NEXT J
        NEXT K
370     REM ** * * * * * ************************** * * * * * **
            *      CHECK TO SEE IF FINISHED THE PROGRAM         *
            ** * * * * * ************************** * * * * * **
380     PRINT MOV$ + "16H",,,,,,,ERASE_LINE;MOV$ + "15H"
        INPUT "       DO YOU WANT TO MAKE ANOTHER SET OF FIELD SHEETS ( Y OR N ) ";ANS$
        SELECT ANS$
          CASE = "Y" , "YES"
            GOTO 130
          CASE = "N" , "NO"
            GOTO PROGRAM_END
          CASE ELSE
            GOTO 380
        END SELECT
400   REM ** * * * * * ****************** * * * * * **
          *            END OF MAIN PROGRAM            *
          ** * * * * * ****************** * * * * * **
```

```
2000     REM ** * * * * * ************************** * * * * * **
             *                ADD_IN_FIELD_VALUES                *
             *   THIS ROUTINE ADDS "__" FOR THE VARIABLE FIELDS *
             * ON THE PRINTED SHEET.                             *
             ** * * * * * ************************** * * * * * **
2005     ADD_IN_FIELD_VALUES:
2010       WQSTN = INPUT_TABLE(J,1)
         STNDES = INPUT_TABLE(J,2)
         PSAM$ = "_________"
         DAY$ = "___"
         MON$ = "___"
         YR$ = "___"
         HR$ = "___"
         MIN$ = "___"
         AST$ = "___"
         LDY$ = "___"
         LDR$ = "___"
         LMIN$ = "___"
         PROC$ = "SAMPLED"
         SUBM$ = "_______"
       RETURN
2500     REM ** * * * * * ************************** * * * * * **
           *                ADD_IN_DATA_VALUES                 *
           *    THIS ASSIGNS WHAT VALUES ARE PRESENT ON THE    *
           * "BIODAT.NEW" FILE TO THE VARIABLES ON THE CONTR   *
           * SHEET.                                            *
           ** * * * * * ************************** * * * * * **
2510 ADD_IN_DATA_VALUES:
         RESET #4
2520     GET #5,KEY #0 EQ VAL%(INPUT_TABLE(J,1))
         PSAM$ = STR$(SAMPLE)
         DAY$ = " " + SEG$(SDATE,1,2) + " "
         MON$ = SEG$(SDATE,4,6) +" "
         YR$ = SEG$(SDATE,8,11)
         HR$ = " " + SEG$(STIME,1,2) + " "
         MIN$ = " " + SEG$(STIME,3,4) + " "
         AST$ = TZONE
         LDY$ = "___"
         LMIN$ = "___"
         LDR$ = "___"
         PROC$ = "TESTED"
         SUBM$ = SUBID
2530     GET #4, KEY #0 EQ STATION
2540   RETURN
4000     REM ** * * * * * ************************** * * * * * **
             *                PRINTING OF THE SHEETS             *
             ** * * * * * ************************** * * * * * **
4003   PRINT_SHEETS:
4005     TITLE_TAB = (109 - LEN(MESS$))/ 2
           PRINT #2,CHR$(12)
           PRINT #2 \ PRINT #2
           PRINT #2, TAB(TITLE_TAB);" B I O L O G I C A L   " + MESS$
           PRINT #2, TAB(TITLE_TAB);" -------------------   " + SEG$(UNDER_LINE,1,LEN(MESS$))
           PRINT #2 \ PRINT #2
           PRINT #2,TAB(10);"STATION # ";WQSTN;TAB(80),"SAMPLE # ";PSAM$
           PRINT #2,TAB(40);STNDES
           PRINT #2 \ PRINT #2
           PRINT #2,TAB(30);"  DATE OF SAMPLING:   DAY: ";DAY$;" MONTH: ";MON$;" YEAR: ";YR$
           PRINT #2 \ PRINT #2
           PRINT #2,TAB(30);"  TIME OF SAMPLING:   HOUR: ";HR$;" MINUTES: ";MIN$;" ZONE: ";AST$
           PRINT #2 \ PRINT #2
           PRINT #2,TAB(30);"  LENGTH OF TIME SAMPLING:  DAYS: ";LDY$;" HOURS: ";LDR$;" MINUTES: ";L
           PRINT #2 \ PRINT #2
```

```
          PRINT #2,TAB(10);"  SUBMITTER ID ";SUBM$;TAB(80);PROC$;" BY _____"
          PRINT #2 \ PRINT #2
4010      REM ** * * * * * ****************************** * * * * * **
              *   DEPENDING ON THE TYPE OF SHEETS WANTED A        *
              *   DIFFERENT DATA TITLE IS PRINTED                 *
              ** * * * * * ****************************** * * * * * **
4020      IF RESP = 1 THEN
            PRINT #2,TAB(14);"SPECIES";TAB(39);"SPECIES";TAB(56);"ID"; &
            TAB(70);"PARAMETER";TAB(86);"PARAMETER";TAB(101);"FIELD METHOD"; &
            TAB(117);"FIELD METHOD"
            PRINT #2,TAB(41);"CODE";TAB(56);"#";TAB(73);"CODE";TAB(89);"VALUE"; &
            TAB(105);"CODE";TAB(121);"VALUE"
          ELSE
            PRINT #2,TAB(14);"SPECIES";TAB(39);"SPECIES";TAB(56);"ID"; &
            TAB(70);"PARAMETER";TAB(86);"PARAMETER";TAB(109);"DESCRIPTION"
            PRINT #2,TAB(41);"CODE";TAB(56);"#";TAB(73);"CODE";TAB(89);"VALUE"
          END IF
          REM ** * * * * * ****************************** * * * * * **
              *    PRINT 16 DOTTED LINES UNDER THE TITLE HEADINGS *
              ** * * * * * ****************************** * * * * * **
4025      FOR M = 1 TO 16
            PRINT #2
            IF RESP = 1 THEN
              PRINT #2,TAB(3);WORD_LINE;TAB(36);VALUE_LINE;TAB(51);VALUE_LINE; &
              TAB(68);VALUE_LINE;TAB(84);VALUE_LINE;TAB(100);VALUE_LINE; &
              TAB(116);VALUE_LINE
            ELSE
              PRINT #2,TAB(3);WORD_LINE;TAB(36);VALUE_LINE;TAB(51);VALUE_LINE; &
              TAB(68);VALUE_LINE;TAB(84);VALUE_LINE;TAB(100);WORD_LINE
            END IF
          NEXT M
          REM ** * * * * * ****************************** * * * * * **
              *  PRINT ( AT THE BOTTOM OF THE SHEET ) UNDERLINED  *
              * LINES                                             *
              ** * * * * * ****************************** * * * * * **
4030      PRINT #2 \ PRINT #2
          PRINT #2,TAB(50);"ADDITIONAL INFORMATION _____"
          PRINT #2
          PRINT #2,TAB(50);"_____"
          PRINT #2
          PRINT #2,TAB(50);"_____"
        RETURN
5000    REM ** * * * * * ************************************** * * * * * **
          *                        INPUT_ROUTINE                        *
          *      THIS ROUTINE WILL COLLECT THE ANSWERS FROM QUESTIONS   *
          *  AND PERFORM ERROR CHECKING.                                *
          ** * * * * * ************************************** * * * * * **
5010    INPUT_ROUTINE:
5020      I = 0
5030      PRINT MOV$ + "16H",,,,,,,ERASE_LINE;MOV$ + "15H"
5040      SELECT RESP
            CASE = 1
              INPUT "            PLEASE ENTER A STATION NUMBER ( TO EXIT '$' )";ANS$
            CASE = 2
              INPUT "            PLEASE ENTER A SAMPLE NUMBER ( TO EXIT '$' )";ANS$
            CASE ELSE
              GOTO PROGRAM_END
          END SELECT
5045      IF ANS$ = "$" THEN
            GOTO END_INPUT
          END IF
5050      IF LEN(ANS$) ) STRG_LEN THEN
            GOSUB PRINT_ERROR
```

```
        END IF
5060    IF I <> 0 THEN
          ANS$ = SEG$(INPUT_TABLE(I,1),1,STRG_LEN - LEN(ANS$)) + ANS$
        END IF
5070    IF LEN(ANS$) <> STRG_LEN THEN
           GOSUB PRINT_ERROR
        END IF
5080    I = I + 1
        INPUT_TABLE(I,1) = ANS$
5085    REM ** * * * * * ****************************** * * * * * **
            *          CHECK THE INPUT DATA FOR ERRORS          *
            ** * * * * * ****************************** * * * * * **
5090    SELECT RESP
          CASE = 1
            RESET #4
            GET #4,KEY #0 EQ ANS$
            INPUT_TABLE(I,2) = STNDES
          CASE = 2
            RESET #5
            ANS = VAL%(ANS$)
            GET #5,KEY #0 EQ ANS
        END SELECT
        PRINT MOV$ + "20H";MOV$ + "0J"
        GOTO 5030
5100  END_INPUT:
      RETURN
6000  REM ** * * * * * ************************************** * * * * * **
          *                      PRINT_ERROR                          *
          *         PRINT A MESSAGE AT THE END OF SCREEN -- WAIT AND   *
          *    THEN ASK THE PROPER QUESTION AGAIN                      *
          ** * * * * * ************************************** * * * * * **
6005  PRINT_ERROR:
6020  PRINT MOV$ + "20H";MOV$ + "0J"
6030  PRINT "     THIS ";MESS1$;"NUMBER ";ANS$;" HAS BEEN INCORRECTLY ENTERED "
      PRINT "     NOW PLEASE RE-ENTER THE ";MESS1$;"NUMBER AGAIN"
      SLEEP 2%
6040  RETURN
30000 REM ** * * * * * ************************************** * * * * * **
          *                     ERROR ROUTINE                         *
          ** * * * * * ************************************** * * * * * **
30010 SELECT ERR
        CASE = 52
          SELECT ERL
            CASE = 200
              RESUME 180
            CASE = 330
              RESUME 310
            CASE = 5090
              GOSUB PRINT_ERROR
              I = I - 1
              RESUME 5030
            CASE ELSE
              RESUME 31011
          END SELECT
        CASE = 155
          SELECT ERL
            CASE = 2530
              STNDES = "NO DESCRIPTION FOUND FOR THIS STATION "
              RESUME 2540
            CASE = 5090
              GOSUB PRINT_ERROR
              I = I - 1
              RESUME 5030
```

```
            CASE ELSE
              RESUME 31011
          END SELECT
        CASE = 145
             GOSUB PRINT_ERROR
             RESUME 5030
        CASE ELSE
          RESUME 31011
      END SELECT
31011 PRINT " ERR ";ERR,"AT LINE ";ERL
31020 PROGRAM_END:
        PRINT #2, CHR$(27) + '?" ' \ CLOSE #2
        PRINT CLRSCREEN
        END
```

```
10 !     ************************************************
   !     *                                              *
   !     *              N A B I O D A T                 *
   !     *                                              *
   !     *      BIOLOGICAL DATA MANAGEMENT SYSTEM       *
   !     *                                              *
   !     *           P R T S P E C . B A S              *
   !     *                                              *
   !     *  THIS PROGRAM PRINTS A FORMATTED LISTING OF  *
   !     * THE FILE SPECIES.NEW . THE USER IS GIVEN     *
   !     * THE OPTION OF LISTING A TAXONOMIC GROUP      *
   !     * WITH THE OUTPUT SORTED BY CODE DESCRIPTION   *
   !     * OR LISTING THE COMPLETE FILE WITH THE        *
   !     * OUTPUT IN ASCENDING CODE ORDER.              *
   !     *                                              *
   !     ************************************************
15 ONERROR GOTO 1000
   !     ****************************
   !     *   MAP AND OPEN FILES     *
   !     ****************************
20 MAP (RECONE) STRING SCODE=6%,DES=4%,DES1=36%,REFER=40%
       MAP (RECONE) STRING SCODE=6%,DESCR=40%,REFER=40%
30 OPEN"[LABS11.BIO]SPECIES.NEW" AS FILE #1,&
       ORGANIZATION INDEXED FIXED,&
       PRIMARY KEY SCODE,&
       ALTERNATE KEY DES DUPLICATES,&
       ALLOW MODIFY,MAP RECONE, ACCESS MODIFY
35 OPEN "LPA0:" FOR OUTPUT AS FILE #2
       PRINT #2, CHR$(27)+'?"#'!          =======) SET PRINTER CHARACTER SIZE
    !  ****************************************
    !  *   DECLARE AND INITIALIZE VARIABLES   *
    !  ****************************************
40 DECLARE STRING TITLE,HEADERS,UNLINE,FFEED,LFEED,CLRSCREEN,GROUP1,LIMIT,&
       BELL,GDES
45 DECLARE INTEGER LINEC,PAGE_NUM,CHOICE
50 TITLE="S P E C I E S    D E S C R I P T I O N    F I L E"
       HEADERS="         SPECIES CODE                                    DESCRIPTION"&
       +"                                    REFERENCE"
       UNLINE= "*************************************************************"&
       +"*************************************************************"
       PAGE_NUM =1
       FFEED=CHR$(12)
       DDATE$=DATE$(0)
       CLRSCREEN=CHR$(27)+"[H"+CHR$(27)+"[2J"
       LFEED=CHR$(10)
       BELL=CHR$(7)
60 PRINT CLRSCREEN;LFEED;LFEED
       PRINT "          1) LIST COMPLETE FILE 'SPECIES.NEW'"
       PRINT "          2) LIST A MAJOR TAXONOMIC GROUP"
       PRINT LFEED
       INPUT "    CHOICE: ";CHOICE
       IF CHOICE(1 OR CHOICE)2 THEN
          PRINT BELL;"INVALID RESPONSE "
          SLEEP 3
          GOTO 60
       END IF
65 IF CHOICE=2 THEN GOSUB 300
70 PRINT #2, FFEED
    !  ********************************
    !  *   PRINT TITLE AND HEADERS    *
    !  ********************************
80 PRINT #2 \ PRINT #2
```

```
        PRINT #2, TAB(42);TITLE
        PRINT #2
        IF CHOICE=2 THEN
            PRINT #2,TAB(10);DDATE$;TAB(48);"MAJOR TAXONOMIC GROUP - ";GDES;&
                TAB(110);"PAGE ";PAGE_NUM
            ELSE
            PRINT TAB(10);DDATE$;TAB(110);"PAGE ";PAGE_NUM
        END IF
        PRINT #2,\ PRINT #2
        PRINT #2,HEADERS
        PRINT #2,UNLINE
        PRINT #2
        LINEC=10
        RESET #1%
85 IF CHOICE=2 THEN 150
        END IF
    !   ********************************************
    !   *  SEQUENTIALLY READ THROUGH SPECIES.NEW   *
    !   *  AND PRINTOUT DATA FOR CHOICE 1          *
    !   ********************************************
90 WHILE 1=1
        GET #1%
        PRINT #2,TAB(10);"    ";SCODE;"                        ";DESCR;"           ";REFER
        LINEC=LINEC+1
        IF LINEC=60 THEN
          GOSUB 200
        END IF
    NEXT
100 GOTO 2000
    !   ***************************************
    !   * PRINTOUT SORTED DATA FOR CHOICE 2 *
    !   ***************************************
150 MAP(RECFOR) STRING SC=6%,SDES=40%,SREF=40%
        OPEN "SORTSPEC.TMP" AS FILE #4%,&
        ORGANIZATION SEQUENTIAL FIXED,&
        ACCESS MODIFY,MAP RECFOR
155 WHILE 1=1
        GET #4
        PRINT #2,TAB(10);"    ";SC;"                       ";SDES;"              ";SREF
        LINEC=LINEC+1
        IF LINEC=60 THEN GOSUB 200 \ END IF
    NEXT
160 GOTO 2000
199 !   ***************************
    !   *  ROUTINE FOR NEW PAGE   *
    !   ***************************
200 PRINT #2,FFEED
        PAGE_NUM=PAGE_NUM+1
        PRINT #2, \ PRINT #2
        IF CHOICE=2 THEN
           PRINT #2,TAB(10);DDATE$;TAB(48);"MAJOR TAXONOMIC GROUP - ";GDES;&
                TAB(110);"PAGE ";PAGE_NUM
         ELSE
           PRINT #2,TAB(10);DDATE$;TAB(110);"PAGE ";PAGE_NUM
        END IF
        PRINT #2
        PRINT #2,HEADERS
        PRINT #2,UNLINE
        PRINT #2
        LINEC=7
    RETURN
299     ! ********************************************************************
        ! * SUBROUTINE FOR SELECTING AND SORTING CODES FROM SPECIES.NEW *
```

```
        ! ********************************************************************
300 PRINT CLRSCREEN;LFEED;LFEED
        INPUT "        ENTER MAJOR TAXONOMIC GROUP 'ONE CHARACTER' ";GROUP1
        IF LEN(GROUP1))1 THEN
           PRINT BELL;"INVALID RESPONSE"
           SLEEP 3
           GOTO 300
        END IF
310 LIMIT=GROUP1+"ZZZZZ"
        GROUP1=GROUP1+"-----"
        ! ************************************
        ! * SELECT DATA FROM SPECIES.NEW AND *
        ! * TO SPECIES.TMP FOR SORTING.      *
        ! ************************************
320 OPEN"SPECIES.TMP" AS FILE #3%,&
        ORGANIZATION SEQUENTIAL FIXED,&
        ACCESS MODIFY,MAP RECONE
325 GET #1,KEY #0 EQ GROUP1
        GDES=DESCR
330 WHILE SCODE( LIMIT
        IF SCODE=GROUP1 THEN 335 \ END IF
        PRINT #3,SCODE;DESCR;REFER
335     GET #1
    NEXT
340 CLOSE #3
        INFILE$="SPECIES.TMP"
        OUTFILE$="SORTSPEC.TMP"
        ! ************************************************************
        ! * INITIALIZE INFO FOR EXTERNAL FILE SORTING FUNCTION *
        ! ************************************************************
350 EXTERNAL INTEGER FUNCTION SOR$PASS_FILES
        EXTERNAL INTEGER FUNCTION SOR$INIT_SORT
        EXTERNAL INTEGER FUNCTION SOR$SORT_MERGE
        EXTERNAL INTEGER FUNCTION SOR$END_SORT
360 MAP(KEYB) STRING SPECODE=6%,DKEY=40%,REFERS=40%,WORD KEY_BUF(4)
        MAP(KEYB) STRING REC_LEN=86%
370 KEY_BUF(0%)=1%
        KEY_BUF(1%)=1%
        KEY_BUF(2%)=0%
        KEY_BUF(3%)=7%
        KEY_BUF(4%)=40%
380 SORT_STATUS%=SOR$PASS_FILES(INFILE$,OUTFILE$)
        IF (SORT_STATUS% AND 1%)=0% THEN
          PRINT "ERROR IN PASS_FILES:";SORT_STATUS%
          GOTO 2000
390 SORT_STATUS%=SOR$INIT_SORT(KEY_BUF(0%),46%)
        IF (SORT_STATUS% AND 1%)=0% THEN
          PRINT "ERROR IN INIT_SORT:";SORT_STATUS%
          GOTO 2000
400 SORT_STATUS%=SOR$SORT_MERGE
        IF (SORT_STATUS% AND 1%)=0% THEN
          PRINT "ERROR IN SORTING FILE:";SORT_STATUS%
        GOTO 2000
410 SORT_STATUS%=SOR$END_SORT
        IF (SORT_STATUS% AND 1%)=0% THEN
          PRINT "ERROR ENDING SORT:";SORT_STATUS%
420 PRINT "                              SORT COMPLETED"
        RETURN

    !   ********************
    !   *  ERROR ROUTINE   *
    !   ********************
1000 IF ERR=11 AND ERL=90 THEN RESUME 2000
```

```
1050 IF ERR=11 AND ERL=155 THEN RESUME 2000
1060 IF ERR=155 AND ERL=325 THEN
        PRINT " THIS CODE NOT ON SPECIES.NEW"
        RESUME 2000
1900 PRINT #2, "ERROR # ";ERR;" IN LINE ";ERL
        RESUME 2000
     !   **********************************
     !   *  CLOSE FILES AND END PROGRAM *
     !   **********************************
2000 CLOSE #1 \ CLOSE #2 \ CLOSE #3 \ CLOSE #4
        KILL "SPECIES.TMP"
        KILL "SORTSPEC.TMP"
        END
```

```
10    !  ***********************************************
      !  *                                             *
      !  *              N A B I O D A T                *
      !  *                                             *
      !  *      BIOLOGICAL DATA MANAGEMENT SYSTEM      *
      !  *                                             *
      !  *           P R T C O D E S . B A S           *
      !  *                                             *
      !  *    THIS PROGRAM WILL PRODUCE A FORMATTED    *
      !  *   LISTING OF BIOCODE.NEW . THE OUTPUT WILL  *
      !  *   BE DIRECTED TO THE PRINTER LPA0: .        *
      !  *                                             *
      !  ***********************************************
15 ONERROR GOTO 3000
      !  *********************************************
      !  *   MAP RECORDS AND INITIALIZE VARIABLES    *
      !  *********************************************
20 MAP (RECONE) STRING BICODE=6%,DES=50%
       OPEN "BIOCODE.NEW" AS FILE #1,&
         ORGANIZATION INDEXED FIXED,&
         PRIMARY KEY BICODE,&
         ACCESS MODIFY,MAP RECONE,ALLOW MODIFY
25 OPEN "LPA0:" FOR OUTPUT AS FILE #2
       PRINT #2,CHR$(27)+'?"#'     !     =============) SET PRINTER CHARACTER SIZE
30 DECLARE STRING TITLE,HEADERS,DDATE,FFEED,UNLINE
       DECLARE INTEGER PAGE_NUM,LINEC
40 TITLE="M E T H O D   A N D   P A R A M E T E R   C O D E   D E S C R I P T I O N   F I L E"
       HEADERS="              CODE                                                    DESCRIPTION"
       UNLINE= "*************************************************************************************************
*****
       PAGE_NUM=1
       FFEED=CHR$(12)
       DDATE=DATE$(0)
     !   *********************************
     !   *   PRINT TITLE AND HEADERS     *
     !   *********************************
50 PRINT #2,FFEED \ PRINT #2 \ PRINT #2
       PRINT #2,TAB(24);TITLE
       PRINT #2
       PRINT #2,TAB(10);DDATE;TAB(110);"PAGE";PAGE_NUM
       PRINT #2 \ PRINT #2
       PRINT #2,HEADERS
       PRINT #2,UNLINE
       PRINT #2
       LINEC=10
     !   *********************************
     !   *   READ THROUGH BIOCODE.NEW    *
     !   *********************************
60 RESET #1%
       WHILE 1=1
          GET #1%
          PRINT #2,"              ";BICODE;"                                          ";DES
          LINEC=LINEC+1
          IF LINEC=60 THEN
             GOSUB 200
          END IF
       NEXT
70 GOTO 3500
     !   *********************************
     !   *   PRINT TOP OF NEW PAGE     *
     !   *********************************
200 PRINT #2, FFEED
       PRINT #2 \ PRINT #2
       PAGE_NUM=PAGE_NUM+1
```

```
        PRINT #2,TAB(10);DDATE;TAB(110);"PAGE";PAGE_NUM
        PRINT #2\ PRINT #2
        PRINT #2,HEADERS
        PRINT #2,UNLINE
        LINEC=7
    RETURN
      !   **********************
      !   *   ERROR ROUTINE    *
      !   **********************
3000 IF ERR=11 AND ERL=60 THEN RESUME 3500 \ END IF
        PRINT #2,"ERROR #";ERR;" IN LINE ";ERL
3500 CLOSE #1 \ CLOSE #2
       END
```